战胜 MATLAB 必做练习 50 题

满晓宇　罗　捷　编著

北 京 大 学 出 版 社

北　　京

内容简介

本书以单元练习的形式，从 MATLAB 最基本的问题入手，循序渐进，逐渐过渡到较为复杂的数学问题、信号分析问题、力学问题和电学问题的求解，将 MATLAB 的学习贯穿在解决不同领域实际问题的过程当中。每一个练习都结合问题，介绍与之相关的 MATLAB 使用知识。全书 50 个练习基本上涵盖了 MATLAB 的主要功能。

本书不仅是一本初学者可以用来入门的教程，而且对于专业设计人员来说，也是一本内容翔实、可供借鉴的参考书。

图书在版编目（CIP）数数

战胜 MATLAB 必做练习 50 题/满晓宇，罗捷编著. 一北京：北京大学出版社，2001.11
ISBN 7-301-05307-X
I.战… II.①满…②罗… III.计算机辅助计算一软件包，MATLAB IV.TP391.75

中国数本图书馆 CIP 数据核字（2001）第 076916 号

书　　名：战胜 MATLAB 必做练习 50 题
著作责任者：满晓宇　罗　捷
责 任 编 辑：黄庆生　汉　明
标 准 书 号：ISBN 7-301-05307-X/TP·0628
出　版　者：北京大学出版社
地　　址：北京市海淀区中关村北京大学校内　100871
电　　话：编辑部（010）62765013　发行部（010）62750672
网　　址：http://cbs.pku.edu.cn
电 子 信 箱：xxjs@pup.pku.edu.cn
印　刷　者：河北省滦县印刷厂
发　行　者：北京大学出版社
经　销　者：新华书店
787 毫米×1092 毫米　16 开本　14.5 印张　371 千字
2001 年 11 月第 1 版　2001 年 11 月第 1 次印刷
定　　价：22 元

前　言

MATLAB 是 Mathworks 公司于 20 世纪 80 年代推出的数值计算软件，近些年来得到了广泛的应用。MATLAB 的全称是 Matrix Laboratory，意思是矩阵实验室。它是以矩阵运算为基础的新一代程序语言。

与 Fortran 和 C 相比，MATLAB 语句显得简单、明了，更加符合人们平常的思维习惯。同时，MATLAB 有着良好的数据可视化功值，能将数字结果以图形的方式表现出来，让人们一目了然。这些特点使得 MATLAB 从众多数值计算语言中脱颖而出。有人称 MATLAB 为第四代计算机语言，它正以相当快的速度在科学研究和工程计算中得到应用和普及。

MATLAB 包括两个部分：基本部分和扩展部分。基本部分主要是指它的数值计算和数据可视化功能；扩展部分则主要是指工具箱。

MATLAB 有着非常强大的数值计算能力，它以矩阵为基本单位进行计算，数域扩展到复数，这一特点决定了 MATLAB 有着非凡的解决数值问题的能力。绘图方面，MATLAB 的绘图语句简单明了、功能齐全。它能够在不同坐标系里绘制二维、三维图形，并能够用不同颜色和快型来描绘曲线。MATLAB 的工具箱其实是由大量 MATLAB 基本语句构成的程序的集合。随着 MATLAB 新产品的不断开发，工具箱的内容也越来越丰富。目前由各个领域的专家开发的工具着已经多达几十个。工具着大大方值了专业领域问题的解决。

随着我国科学技术的发展和国内外合作的加强，各学科与计算机技术融合的趋势越来越强。MATLAB 作为一种有代表性的数值计算语言，渐渐被大学生、研究生、科研工作者和工程技术人员所技受和喜爱。目前，在我国，MATLAB 已经得到了一定程度的普及，讲述 MATLAB 使用方法的书籍也不在少数。但是，利用 MATLAB 解决多学科具体问题的参考书却凤毛麟角。本书试图做填补空白的尝试，目的是抛砖引玉。

本书以单元练习的形式，从 MATLAB 最基本的问题入手，循序渐进，逐渐过渡到较为复杂的数学问题、信号分析问题、力学问题和电学问题的求解，将 MATLAB 的学习贯穿在解决不同领域实际问题的过程当中。每一个练习都结合问题，介绍与之相关的 MATLAB 快用知识，全书 50 个练习基本上涵盖了 MATLAB 的主要功能。本书的练习 1 至练习 15 主要介绍 MATLAB 的基础知识和一般操作；练习 16 至练习 31 是具体数学问题的求解；练习 32 至练习 40 主要介绍了信号分析问题求解和偏微分方程工具箱的使用；练习 41 至练习 45 是常见力学问题的求解；练习 46 至练习 50 介绍了电学问题求解。

本书由满晓宇、罗捷主编，另外，余晓鹏、张伟华、何广、张石勇、战祥森、张松伟、吴绍伟、孙科峰、渠值永、覃文圣、牟南、钟光辉、钱辰、王晓龙、邓瑞峰、肖健、解灵运等也参加了本书编写工作。

编　者

2001 年 11 月

目　　录

练习 1　基本操作和简单语句输入

知识背景

MATLAB 之所以能够成为优秀的数学软件之一，得益于其强大的数值计算能力和近乎完美的数据可视化功能。在数值计算方面，MATLAB 相对于其他常用的数值计算语言，比如 Fortran 和 C，很大的优点在于其语句的简洁性和易沟通性。对于 Fortran 或 C 的使用者来说，应付繁琐的语句和冗长的代码始终是一件让人头痛的事，而 MATLAB 语句会使你有耳目一新的感优。MATLAB 的这一特点再加上它具有的友好界面使得 MATLAB 在众多数学软件中脱颖而出。下面我们就从简单操作和基本的语句输入开始，去领略一下 MATLAB 的魅力。

主要内容

【本练习考查知识点】

本练习将要介绍 MATLAB 一些基本操作和简单语句函值的输入及相关功输的实现。这一部分的内容虽然简单，但它是熟练掌握 MATLAB 的第一步。俗话说：有了良好的开端就等于成功了一半。相信读者能够通过本练习达到对这一部分内容驾轻就熟的程度。

练习过程

（1）我们首先来认识 MATLAB 的命令窗口。在 Windows 下开始运行 MATLAB，就会出现如图 1-1 所示的界面，最上面显示"MATLAB Command Window"字样的高亮条部分叫值标题栏，它表明当前窗口是命令窗口。标题栏最右边的三个按钮依次为窗口最小化、窗口缩放和窗口关闭按钮。标题栏下面是菜单栏，它包含"File（文件）"、"Edit（编辑）"、"View（查看）"、"Window（窗口）"、和"Help（帮助）"五个选项。菜单栏下面有 10 个工具值值，将鼠标移到上面将会显示出相应按钮的功能，读者可根据提示进行操作。在以后的练习中我们会逐渐熟悉这些功能。

工具栏下面的大片区域是命令值入区，MATLAB 的命令就是从这里输入的。读者不难发现每次打开 MATLAB 命令窗口时，都会在输入区内显示下面的信息：

To get started, type one of these: helpwin, helpdesk, or demo.

For product information, visit www.mathworks.com.

上面的文字说明 MATLAB 的使用者可以通过在命令区里键入 helpwin，helpdesk 而得到联机帮助，键入 demo 可以看到 MATLAB 自带的演示实例。(如图 1-2、图 1-3 和图 1-4 所示）读者可以从中找到大多数自己想要了解的内容。如果读者想进一步了解产品信息，可以登录相关网站。

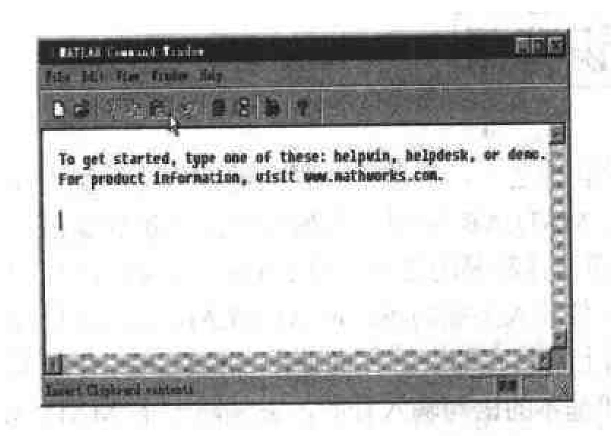


图 1-1　MATLAB 命令窗口



图 1-2　Helpwin 帮助窗口

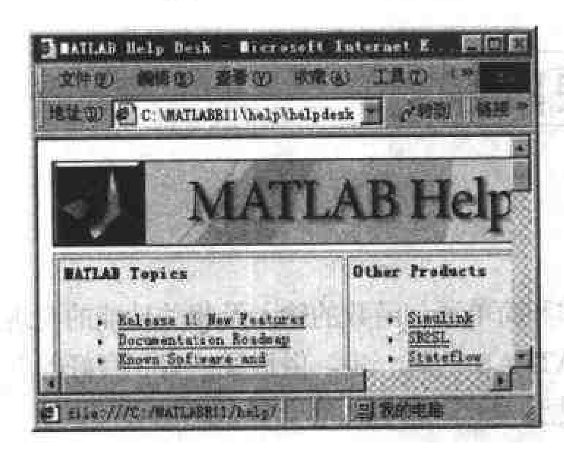


图 1-3　Helpwin 帮助窗口

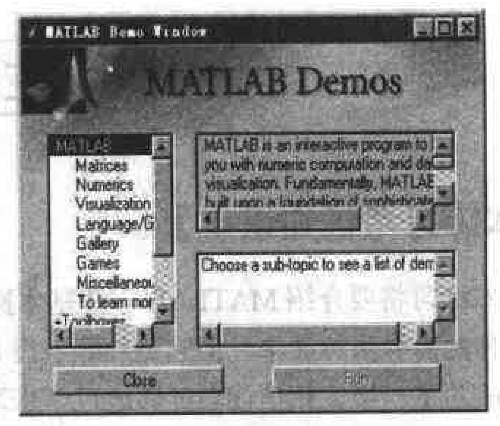


图 1-4　MATLAB 演示窗口

(2) 下面我们来加深对命令窗口的了解。单击菜单栏上的"File"选项，就会弹出一个下拉菜单（如图 1-5 所示)，请读者练习下列操作。

- "New"：单击"New"选项，你会看到一个包含三项内容的右拉式子菜单。单击"M-file"，就会打开指定的编辑器，并且随之产生一个空白的 M 文件。M 文件是包含 MATLAB 源代码的文本文件。MATLAB 自带了大量 M 文件，当然你也可以在空白 M 文件中编制自己想要的源代码文件，这在 MATLAB 数值计算中是一项很重要的功能。
- "Open"：单击"Open"选项，会弹出一个对话框，要求你输入想要打开的文件，可以是 M 或 Fig 文件，也可以是其他可用类型的有效文件。
- "Open Selection"：通常情况下这个选项呈现灰色，只有当你在命令窗口的编辑区内选中一个 M 文件时，才变成可用。单击此选项将会用特定编辑器打开被选中的 M 文件。
- "Run Script"：单击"Run Script"选项，在对话框内输入或选中可执行的目标文件。

单击“OK”，MATLAB 将会执行被选中的目标文件。

- “Load Workspace”：单击“Load Workspace”选项，会弹出一个对话框，要求选定想要打开的文件。这里要求文件为 Mat 文件（*.Mat 是 Workspace 里的变量存储文件），打开后将会把文件中保存的变量载入当前的工作空间中。
- “Save Worksapce As”：单击“Save Worksapce As”选项，在弹出的对话框里填入一个文件名，就会把当前工作空间里的变量以“.Mat”形式存储起来。你可以自己设置存储路径，文件名就是刚才填入的那一个名字。
- “Show Workspace”：单击“Show Workspace”选项，你会看到名称为“Workspace Browser”的界面，这就是工作空间浏览器，在这里你可以看到当前工作空间里所有的变量信息。
- “Show Graphics Property Editor”：单击“Show Graphics Property Editor”选项，会弹出图形属性编辑器，在这里你可以对 MATLAB 可视化窗口的属性进行修改。
- “Show GUI Layout Tool”：单击“Show GUI Layout Tool” 选项，MATLAB 会打开图形界面控制面板，你可以利用它绘制自己喜欢的图形界面。
- “Set Path”：单击“Set Path”选项，就会弹出图 1-6 所示的路径浏览器。可以在弹出的路径浏览器中更改 MATLAB 的搜索路径。（在利用 MATLAB 进行数值计算过程中，有时会出现路径设置问题，这时就需要打开“Set Path”，然后按照文件的存储位置对将要调用文件的搜索路径进行更改。）“Path”窗口里显示了 MATLAB 的两个系统默认路径“Toolbox”和“Work”。用鼠标双击“Path”窗口列表里的任意一个分项，右边的“File in general”就会给出该分项所包含文件的信息。

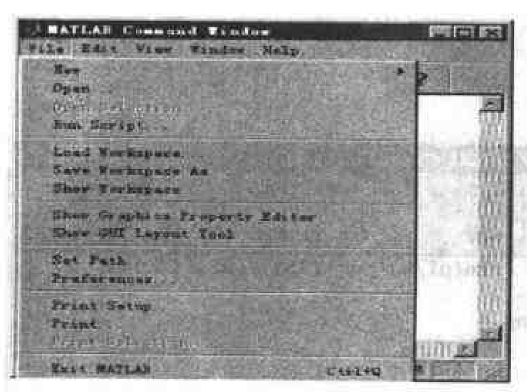




图 1-5　“File”下拉菜单

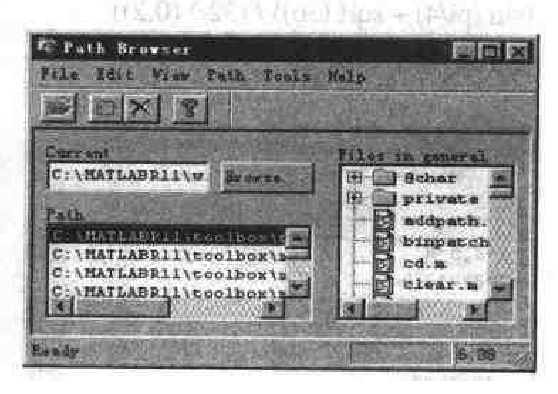


图 1-6　路径浏览器

- “Preferences”：单击“Preferences”选项，打开命令窗口的显示格式选项卡，通过对选修卡的设置，可以改变命令编辑区的显示格式。
- “Print Setup”：“Print Setup”选项在未安装打印机前呈灰色，即不可用。单击，并按照提示安装好打印机后，可以通过此项设置打印机参数。
- “Print”：“Print”选项在未安装打印机前呈灰色，安装打印机后，单击此项，即可进行打印。
- “Exit MATLAB”：单击“Exit MATLAB”或者使用快捷键【Ctrl+Q】，将退出 MATLAB。

（3）下面我们练习简单语句输入。

计算　$y=a\times b+a\div b$　（$a=3$，$b=4$）　的值

MATLAB 语句的一般形式为：

变量＝表达式

在 MATLAB 命令窗口里输入：

a=3;b=4;

y=a*b+a/b

得到的输出结果如图 1-7 所示。

如果变量和“＝”省略，MATLAB 会自动建立名为“ans”的变量。输入语句后以回车结束，就会在工作区里显示计算结果。对初学者来说，特别值得注意的是，如果输入的语句以“;”结束，MATLAB 则只进行计算而不输出结果；如果以“,”结束，MATLAB 会输出计算结果。

在某些变量很多，但只需要知道最终结果的情况下，应该注意合理使用“;”，否则输出的结果将比较乱，且会影响运行速度。

MATLAB 的变量由字母、数字和下划线组成，最多可以有 31 个字符，第一个字符必须是字母。在这里还有一个值得引起注意的地方：MATLAB 的变量是要区分大小写的，这一点初学者往往会忽视。要建立一个新的变量，只需要输入变量名，MATLAB 会自动为其建立变量。

我们再来看一个例子：

计算 $\dfrac{\sin 45^\circ+\sqrt{36}}{\sqrt[5]{32}}$

在 MATLAB 命令窗口内输入：

(sin (pi/4) + sqrt (36)) / (32.^ (0.2))

得到的输出结果如图 1-8 所示。

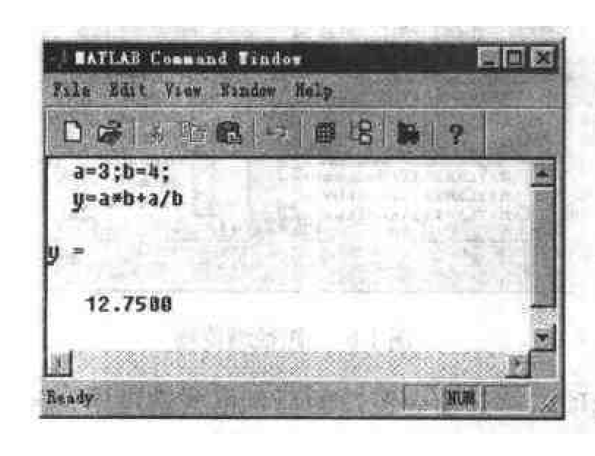


图 1-7　输出结果

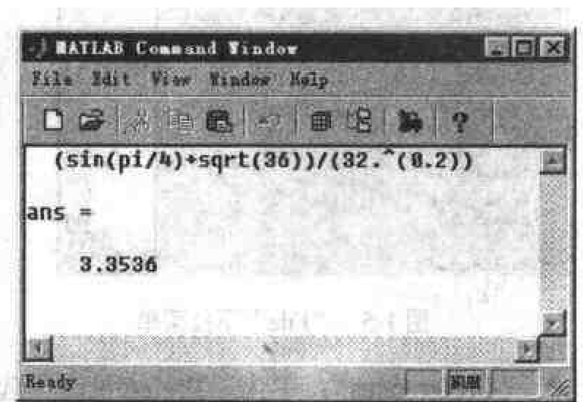


图 1-8　输出结果

读者还需知道的是：MATLAB 在行尾加上“…”表示续行。运算符前后的空格不影响计算结果。这些较为细小的方面也应该引起注意。

从以上两个简单例子我们可以看，MATLAB 的语句确实简洁明了，界面也很友好。当然这只是最基本的例子，在后面的练习我们要逐渐接触较为复杂的 MATLAB 语句。

【练习小结】

本练习作为整本书的第一个练习，旨在引导读者对 MATLAB 这一数学软件有一个宏观的印象，使读者初步了整 MATLAB 命令窗口的功数，能够在命令窗口中输入简单语句，且得到运行结果。

本练习的内容虽然简单，但却很重整，特别是语句输入中“,”与“;”的用法区别，大小写字母代表不同变量等整点，请读者牢记。

【思考题】

1．请打开 MATLAB 命令窗口，调出关于“MATLAB\General”的帮助文件。
2．请调出 MATLAB\Stateflow 的演示实例。
3．计算：

$y = x^3 + (x-0.98)^2/(x+1.25)^3 - 5(x+\frac{1}{x})$，$x=2, x=4$ 时的值。

4．计算：

$\cos 60^\circ - \sqrt[3]{9-\sqrt{2}}$

5．已知：$a=3, A=4, b=a^2, B=b^2-1, c=a+A-2B, C=a+B+2c$

求：C

练习 2　矩阵和数组的一般操作

知识背景

在线性代数中，我们经常用到数组和矩阵这两个概念。我们先来定义矩阵：

$$\begin{bmatrix} a_{11} & a_{12} & \Lambda & a_{1n} \\ a_{21} & a_{22} & \Lambda & a_{2n} \\ \Lambda & \Lambda & \Lambda & \Lambda \\ a_{m1} & a_{m2} & \Lambda & a_{mn} \end{bmatrix}$$ 称为 $m \times n$ 阶矩阵，记为 $(a_{ij})_{m\times n}$ 或 $A_{m\times n}$

若矩阵 A 与 B 的行数与列数均相同，则称 A 与 B 为同型矩阵。

我们所说的数组是一种特殊的矩阵。MATLAB 中的数组可以认为是行向量，也就是只有一列的矩阵。数组的运算和矩阵的运算既有相似之处，也有很多不同。请读者认真总结，加以区分。

主要内容

【本练习考查知识点】

矩阵运算是 MATLAB 中最为基本也最为重要的部分。我们在这个练习中将学习关于矩阵和数组的一般操作，比如创建、保存、修改和调用。

练习过程

1. 矩阵的创建

（1）当需要的矩阵操数不大，比较方便的方法是在工作区中直接输入矩阵。矩阵的首尾要以“[ ]”括起来，同一行中的元素之间用“,”或空格隔开，行与行之间用“;”或【Enter】键分开。矩阵的元素可以是数字或表达式。表达式中不能包含没有定义的变量。元素的赋值由表达式完成。

我们来看一个例子：

在 MATLAB 命令区中输入：

x=pi/4；y=2；

A=[3，4，5；sin(x)，y^2，9]

运行结果如图 2-1 所示。

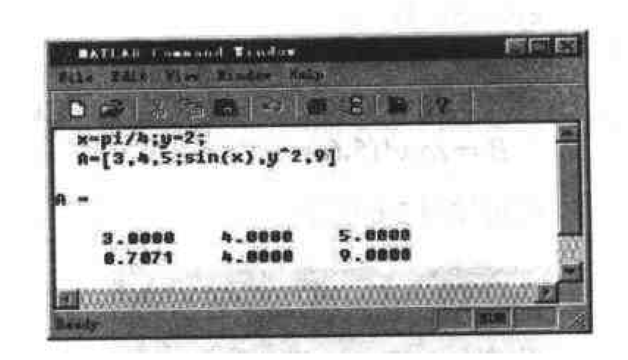


图 2-1　创建矩阵

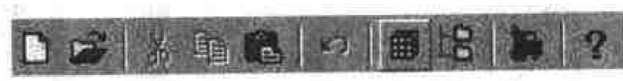
由上例看到，对于合法的表达式，MATLAB 将以表达式的值为元素赋值。

（2）当需要的矩阵很大，不适用于在工作区中直接输入时，可以使用 MATLAB 提供的“Matrix Editor”，即矩阵编辑器来完成输入和修改。在使用矩阵编辑器，必须首先在工作区中定义一个变量。这个变量可以是数或者简单矩阵。请看以下步骤：

在命令区中输入：

A=1;

用鼠标单击工具栏的工作区浏览器，如图 2-2 所示。



Workspace Browser

图 2-2　工具栏

MATLAB 这时弹出变量浏览器，显示出变量的信息，如图 2-3 所示。

选中变量 A，这时就可以打开或删除变量 A。用鼠标左键双击 A，或者单击“Open”按钮，就打开了矩阵编辑器。左下脚的两个文本框分别要求输入希望得到的矩阵的行数和列数。填好维数后，可以用鼠标选中表格中我们想要修改的元素，将原来的元素修改为我们想要的值，从而建立起一个多维矩阵。矩阵编辑器右下角的文本框用来显示当前选中的元素。



图 2-3　变量浏览器

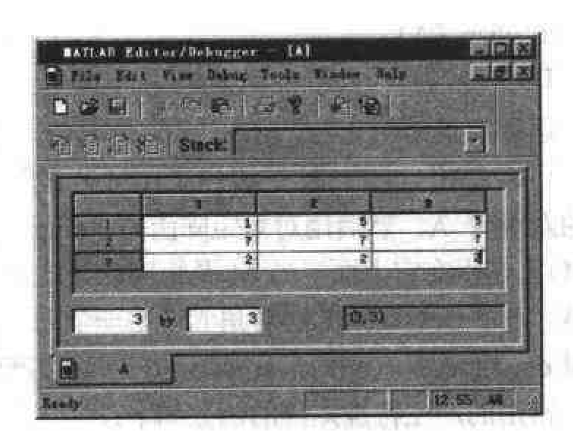


图 2-4　矩阵编辑器

我们应该注意到，在修改元素的值时，可以直接输入数值，也可以输入表达式。修改好后，关闭矩阵编辑器，新的矩阵 A 就保存好了。我们还可以利用矩阵编辑器将原矩阵裁减为它左上方的主子矩阵，也可以把原矩阵扩大为更大的矩阵，MATLAB 系统会自动将扩展部分的元素设置为 0。

（3）对于一些常用的特殊矩阵，比如对角阵、单位阵、零矩阵等常用的矩阵和由它们变换和截取产生的新矩阵，我们可以用 MATLAB 自带的函数来创建它们。

比如我们想建立一个单位阵，可以输入下面的语句：

eye（3，4）

得到的结果如图 2-5 所示。

或者，我们要创立一个随机矩阵，可以输入下面的语句：

$$B = rand(5,6)$$

结果如图 2-6 所示。



图 2-5　创建单位矩阵

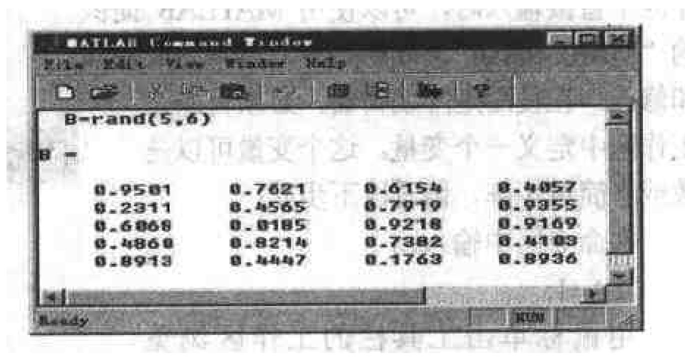


图 2-6　创建随机矩阵

请读者注意，随机矩阵顾名思义，每次的元素数值都是随机产生的，前后两次矩阵不同，不应感到奇怪。

我们再来练习创建一个对角阵：

在命令区中输入：

$C = A$

d＝diag（A）

D＝diag（d）

运行结果如图 2-7 所示。

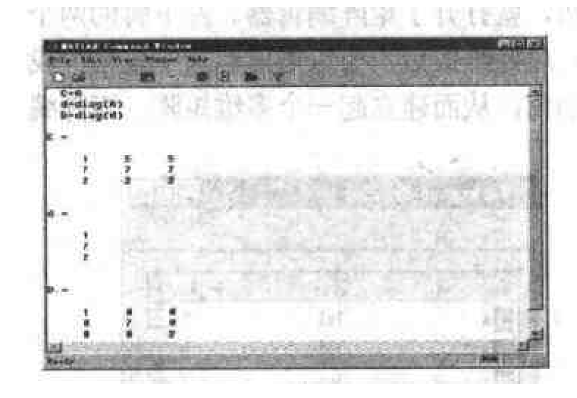


图 2-7　创建对角阵

可以看出，我们首先调出了刚才修改过的新矩阵 A，然后用对角矩阵函数“diag（M）”作用在它上面。当 M 是矩阵时，比如 A，diag 函数取矩阵的对角元产生一个列向量 d；当 M 是向量时，比如 d，diag 函数产生一个对角阵 D。

常用的产生特殊矩阵的函数如下：

- eye（m，n）　　单位阵
- rand（m，n）　　随机矩阵
- randn（m，n）　　正态分布的随机矩阵
- zeros（m，n）　　零矩阵
- ones（m，n）　　全部元素都为 1 的矩阵
- compan（A）　　矩阵 A 的伴随矩阵
- gallery　　测试矩阵
- bankel（m，n）　　n 维 Hankle 矩阵

- invhilb（n）　　n 维逆 Hilbert 矩阵
- magic（n）　　n 维 Magic 方阵
- toeplitz（m，n）　　Toeplitz 矩阵
- wilkinson（n）　　n 维 Wilkinson 特征值测试矩阵
- handamard（n）　　n 维 Handamard 矩阵
- hilb（n）　　n 维 Hilbert 矩阵
- kron（A，B）　　Kronecker 张量积
- pascal（n）　　n 维 Pascal 矩阵
- vander（A）　　由矩阵 A 产生 Vandermonde 矩阵

2. 矩阵的保存和调用

如果想要在今后继续使用已经创建的矩阵，就需要将当前矩阵以文件形式储存起来，以便下次使用时调用。

我们首先来试着保存在前面的例子中创建的矩阵 D。

在命令区中输入：

Save　mydata　D

其中 mydata 为我们给变量文件起的名字。系统会自动沿设定好的路径以“.mat”格式存储文件。如果想将文件储存到别的地址，可以在文件名前加上想要储存的路径。比如：

save　D:\homework　mydata　D

如图 2-8 所示，在系统路径 MATLABR11\work 下出现了 mydata.mat 这个新文件，它就是我们保存的变量文件。

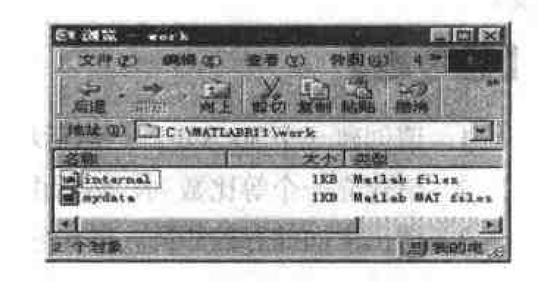


图 2-8　保存结果

如果想调用这个文件，只需要在命令区输入：

load　mydata

就可以将保存在文件中的矩阵读到 MATLAB 工作区的内存中来。

3. 数组的创建和保存

前面说过，数组可以看成是行向量，即只有一列的矩阵。因此前面介绍的关于矩阵的方法对于数组也是适用的。下面介绍一些 MATLAB 中创建特殊数组的命令。

我们输入：

```
a=linspace(0,1,5)
b=logspace(0,4,5)
```

运行结果如图 2-9 所示。

我们看到，“linespace”用于产生一个等差数列，括号里的三项分别表示起始值、终止值和元素数目；“logspace”用于产生一个等比数列，在上面的例子中产生了一个起始值为 10^0，终止值为 10^4，元素数目为 5 的等比数列。

如果输入：

```
a=linspace(0,1,5.5)
b=logspace(0,4,5.5)
```

得到结果如图 2-10 所示。

```
a=linspace(0,1,5)
b=logspace(0,4,5)

a =

        0    0.2500    0.5000    0.7500    1.0000

b =

        1        10       100      1000
```

图 2-9　创建特殊数组

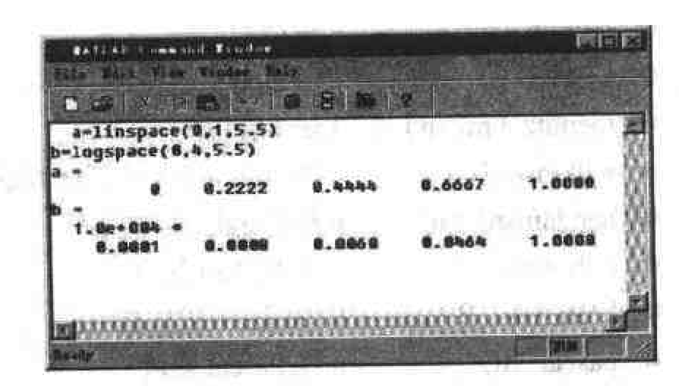


图 2-10　创建特殊数组

请注意当元素数目不是整数时函数的结果。

【练习小结】

本练习主要向读者介绍了 MATLAB 中矩阵和数组的创建、保存、修改和调用。有关于矩阵和数组的命令函数是这一节的重点，希望读者能够在这方面多下点功夫，达到熟练的程度。

【思考题】

1. 请创建一个3×3矩阵，然后利用矩阵编辑器将其扩充为 4×5 矩阵。
2. 请建立一个等比数列，然后由它产生一个对角阵，并储存这个矩阵。
3. 调出上面储存的矩阵，并由它产生一个列向量。

练习 3　矩阵和数组的操作

知识背景

上一个练习中，我们学习了矩阵和数组的创建、保存、修改和调用，但那只是矩阵操作中比较简单的部分。事实上，矩阵的变换是比较复杂的，MATLAB 中有关矩阵变换的命令和函数也比较多，要想掌握好这部分内容，必须进一步学习有关矩阵和数组的操作。

不知大家注意到了没有，在 MATLAB 里矩阵和向量这两个词经常混用。准确的说，矩阵是以实数或复数为元素的长方形向量。在 MATLAB 里，列向量被当作只有一列的矩阵，而行向量被当作只有一行的矩阵。因此对于矩阵适用的函数或命令对数组来说也是适用的。MATLAB 提供了几十个函数命令，用来生成不同矩阵，基本上囊括了我们所能见到的所有矩阵类型。下面我们先来接触 MATLAB 中几个常见的矩阵类型和矩阵变换：

（1）随机矩阵（Random Matrix）：矩阵的元素为 0～1 之间的某个小数，由 MATLAB 随机产生，具有不确定性。

（2）单位阵：主对角线元素为 1，其余元素为 0 的矩阵。

（3）魔方阵（Magic Matrix）：任意行、列及对角线元素之和相等。

（4）转置矩阵：将矩阵第 i 行第 j 列的元素放到第 j 行第 i 列，这样产生的新矩阵叫做原来矩阵的转置矩阵。

（5）子矩阵：从矩阵中提取一部分矩阵，则提取的矩阵称为原矩阵的子矩阵。

主要内容

【本练习考查知识点】

矩阵操作是 MATLAB 中最为基本的内容，也是进行矩阵计算所必需的基本功。在本练习中，我们将继续学习矩阵和数组的操作，以期能够为后面的学习打好坚实的基础。

练习过程

（1）矩阵的修改、扩充、剪切、拼接及 0－1 向量提取。

上一个练习中，我们学习了用矩阵编辑器来改变矩阵中的元素。但如果我们想在工作

区中直接调用或修改矩阵元素，应该怎么办呢？其实很简单。比如我们想要建立一个3×4的随机矩阵 A，然后改变部分元素的值。

在命令区中输入：

A=rand（3，4）

按回车后，就产生了一个3×4的随机矩阵。如果我们想要将矩阵 A 第三行第二列的元素改为 10，可以输入：

A（3，2）=10

按回车后，命令区中就会显示改变后的矩阵，如图 3-1 所示。

如果我们在命令区中输入：

A=rand（3，4）

A（4，5）=10

按回车后，则3×4 的随机矩阵被扩充成了4×5 的矩阵，其中新矩阵第 4 行第五列的元素等于 10。其他扩充后新增的元素系统都设为 0，如图 3-2 所示。

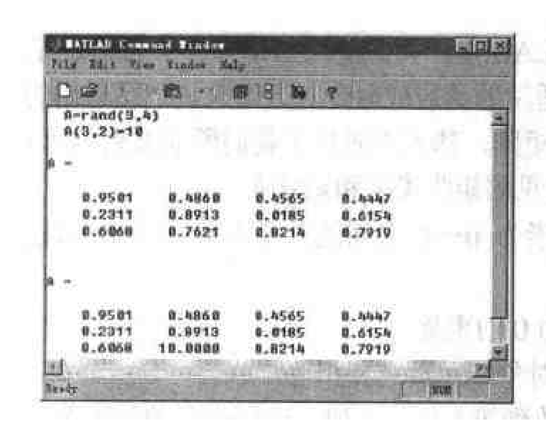


图 3-1　修改矩阵元素

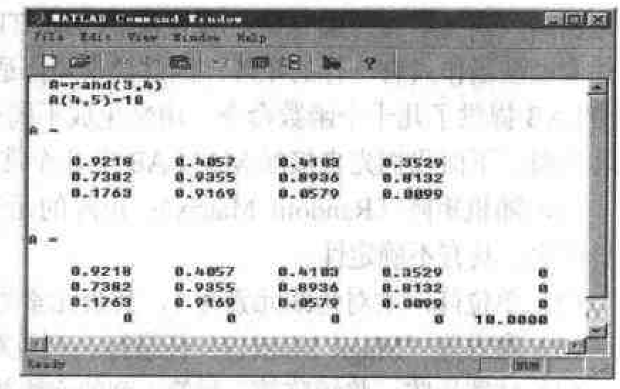


图 3-2　修改矩阵

请读者仔细比较上面两个例子。

再来看一看如何标识矩阵和提取子矩阵。

我们可以用标量、向量、冒号的标识来引用矩阵的子阵并为其赋值。子阵的序号向量标识方式为 A（u，v)，u、v 的值应大于等于 1，并且小于等于矩阵的维数。u、v 中的任何一个可以是“：”，这表示 u 所在的位置为全部行，或 v 所在的位置为全部列。“0-1”向量的标识方式为 A（L_1，：)、A（：，L_2)、A（L_1，L_2)。向量 L_1、L_2 中的元素或取 1（表示提取了相应的行或列）或取 0（表示没有提取)。我们看一个具体的例子。分别在命令区中输入：

```
A = rand (3, 4 )
A1 = A (2:3, [1,3,4] )
A2 = A ([1,3], [2,4] )
```

```
A = rand (3,4 )
L = A (2, :) < 0.5
A3 = A (2,L)
```

得到的结果如图 3-3、图 3-4 所示。

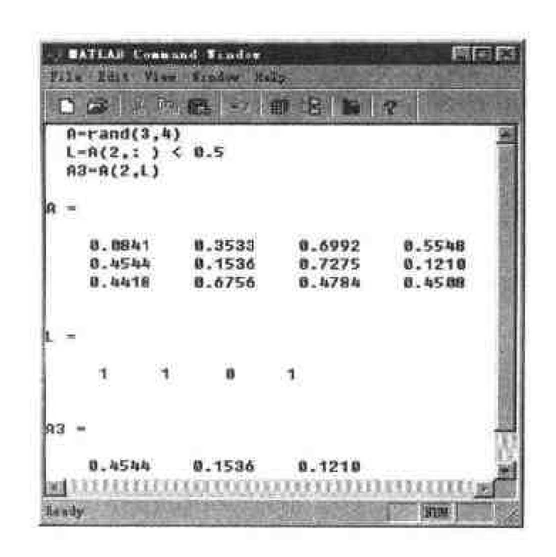


图 3-3　矩阵提取

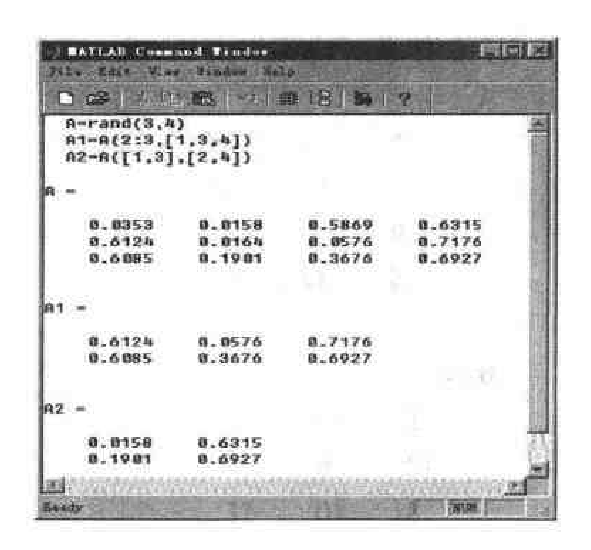


图 3-4　0—1 向量提取

上面的例子主要是从矩阵中提取或称为裁剪矩阵，事实上 MATLAB 也允许拼接矩阵，比如在命令区中输入：

B = [rand (2) , ones (2) ； eye (2 , 4)]

则先将二阶随机矩阵和二阶单位阵进行左右拼接，然后再和 2×4 对角阵进行上下拼接，从而得到一个新矩阵。如图 3-5 所示。

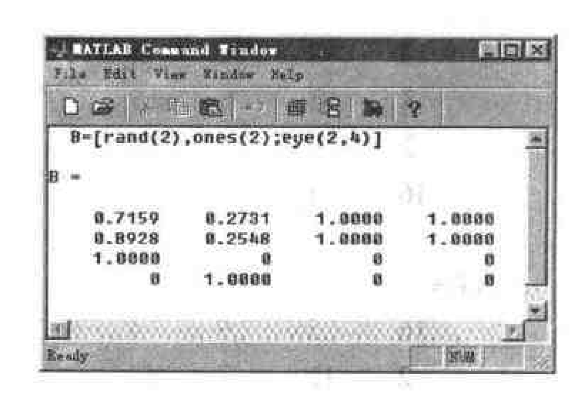


图 3-5　矩阵拼接

（2）通过矩阵的结构变换，获得新矩阵。

MATLAB 中提供了一些函数，利用这些函数通过矩阵的旋转、变维和截取元素来得到新矩阵，具体如表 3-1 所示。

表 3-1　通过矩阵的结构变换，获得新矩阵

函数	说明
L=tril(A)	L 主对角线及以下元素取矩阵 A 的元素，其余为 0
L=tril(A,k)	L 及第 k 条对角线及以下元素取矩阵 A 的元素，其余为 0
U=triu(A)	U 主对角线及以上的元素取矩阵 A 的元素，其余为 0
U=triu(A,k)	U 第 k 条对角线及以上的元素取矩阵 A 的元素，其余为 0
B=rot90(A)	矩阵 A 逆时针旋转 90° 得到 B
B=rot90(A,k),	矩阵 A 逆时针旋转 k×90° 得到 B
B=fliplr(A)	矩阵 A 左右翻转得到 B
B=flipud(A)	矩阵 A 上下翻转得到 B
B=reshape(A,m,n)	将矩阵 A 的元素重新排列，得到 m×n 的新矩阵。(m×n 应等于 A 的行列数之积。若 A 为3×4，则 m，n 可为 2，6 或4，3 等)

请看下面的例子，在命令区中输入：

```
A=magic(4)              B1=rot90(A)
B2=rot90(A,2)           BT=A'
```

得到的结果如下：

```
A  =
     16     2     3    13
      5    11    10     8
      9     7     6    12
      4    14    15     1


B2 =
      1    15    14     4
     12     6     7     9
      8    10    11     5
     13     3     2    16


B1 =
     13     8    12     1
      3    10     6    15
      2    11     7    14
     16     5     9     4


BT =
     16     5     9     4
      2    11     7    14
      3    10     6    15
     13     8    12     1
```

请读者注意矩阵旋转与转置的区别。

在命令区中输入：

A=magic（4）

B=tril(A,2)

C=triu(A,－1)

得到的结果如图 3-6 所示。

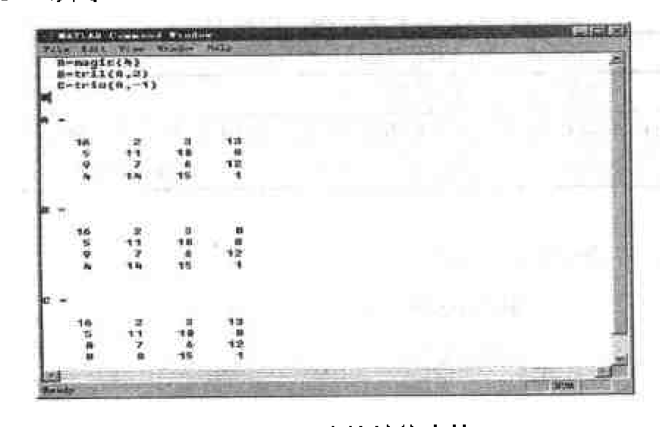


图 3-6　矩阵的结构变换

请注意，当 tril（A，k） 和 triu（A，k）中 k 为正整数值时，表示主对角线上第 k 条角线；k 为负整数时，表示主对角线下第 k 条对角线。

在练习的最后，向大家介绍一下 MATLAB 中编辑命令行时常用的按键功键，使用这些功能，能够实现语句重调和修改（如表 3-2 所示）。

表 3-2　键盘操作

按键操作		作　用
↑	Ctrl+P	调出前一行（调出前面的命令后修改）
↓	Ctrl+N	调出后一行
←	Ctrl+B	光标前移一个字符
→	Ctrl+F	光标后移一个字符
Ctrl+→	Ctrl+R	光标前移一个字
Ctrl+←	Ctrl+L	光标后移一个字
Home	Ctrl+A	光标移动到行首
End	Ctrl+E	光标移动到行尾
Esc	Ctrl+U	清除一行
Del	Ctrl+D	清除光标后的字符
Backspace	Ctrl+H	清除光标前的字符
	Ctrl+K	删除到行尾

【练习小结】

本练习进一步向读者介绍了 MATLAB 中矩阵和键组的创建、修改、和变换，总结了 MATLAB 中的矩阵变换命令和函数及常用操作键与相关功能。其中矩阵变换命令和函数在以后的矩阵计算中要经常用到。

【思考题】

1．创建 3×4 魔方阵（magic matrix）和相应的随机矩阵（random matrix），将两个矩阵拼接起来。然后键取任意两个列向量。

2．调用上题中的拼接矩阵，并求其转置矩阵。

3．创建一个 4×4 单位阵，提取主对角线以上的部分。

4．创建一个 4×5 键机矩阵，提取第一行和第二行中大于 0.3 的元素组成的矩阵。

5．按照表 3-2 练习键盘操作。

练习 4　矩阵和数组的加减运算与乘法

知识背景

在数学计算或工程数据处理中，关于矩阵和数组的计算司空见惯。如果人工去做这部分工作，将既费时又费力并且容易出错。所幸的是 MATLAB 为我们展示了丰富而出色的矩阵运算功能，大大简化了我们的工作量。为了使读者尽快熟悉这部分内容，我们先来回顾一下矩阵计算的基本定义：

（1）若矩阵 A 与 B 的行数与列数相同，则称 A 与 B 为同型矩阵。同型矩阵可进行加减运算，$A \pm B$ 就等于矩阵 A 与 B 对应元素的代数和。

（2）所有元素均为 0 的矩阵，称为零矩阵。

（3）数乘矩阵：设 k 为常数，A 为 $m \times n$ 阶矩阵，则定义数乘矩阵为：

$$kA = \begin{bmatrix} ka_{11} & ka_{12} & ka_{13} & ka_{14} \\ \Lambda & \Lambda & \Lambda & \Lambda \\ ka_{m1} & ka_{m2} & \Lambda & ka_{mn} \end{bmatrix}$$

（4）矩阵的乘法设 A 为 $n \times s$ 阶矩阵，B 为 $s \times m$ 阶矩阵，则定义 C＝AB（其中 C 为 $n \times m$ 阶矩阵）为矩阵 A 与 B 的乘积。

$$C_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \Lambda + a_{is}b_{sj} = \sum_{t=1}^{s} a_{it}b_{tj}$$

主要内容

【本练习考查知识点】

矩阵运算是 MATLAB 中最基本也最为重要的部分。我们在这个练习中将学习矩阵和数组的加减法和乘法。本练习将练习使用矩阵和数组各自的运算符号进行简单计算，使读者体会它们之间的联系与区别。数组的运算符比相应的矩阵运算符多一个小黑点，这一点要牢记。另外我们还将在简单计算中复习以前学过的矩阵的创建和调用。

练习过程

MATLAB 系统提供了如下的矩阵运算符：

＋ 加法　－ 减法　* 乘法　^ 幂　\ 左除　/ 右除　' 转置

使用上述矩阵运算符要符合矩阵运算的要求。比如：只有同型矩阵才能相加减；A*B 必须满足 A 的列数等于 B 的行数；只有方矩阵（行数等于列数）才可以求幂……

（1）矩阵的加减运算

在命令区中输入：

$A = rand\ (3,4)$

$B = rand\ (3,4)$

$C = A + B$

得到的结果如图 4-1 所示。

可见矩阵相加即将两个矩阵的对应元素相加，得到一个新矩阵。同样的，矩阵减法即是将两个同型矩阵的对应元素相减。读者可以计算一下 $A-B$ 的值。

如果计算矩阵和标量的和，在命令区中输入：

A = rand（3,4）

B = A+3

得到的结果如图 4-2 所示。

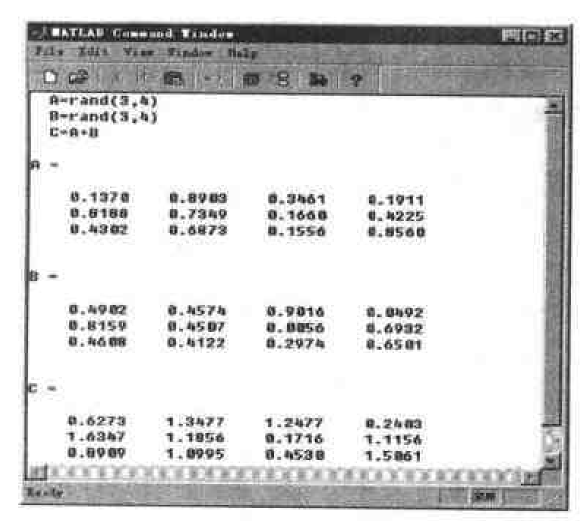


图 4-1　矩阵相加

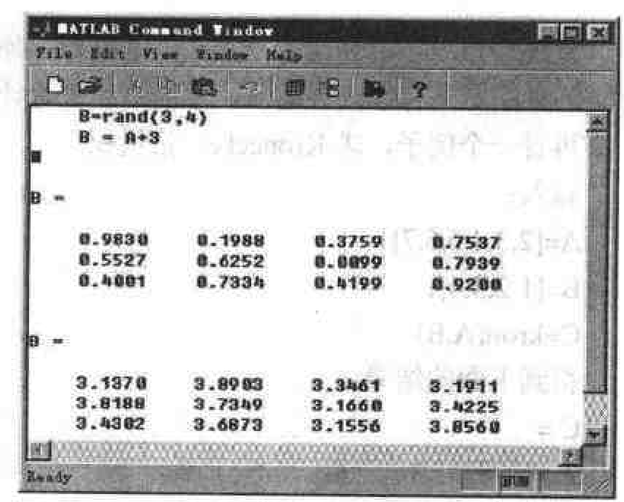


图 4-2　矩阵与标量相加

由图 4-2 知，标量与矩阵（1×1 矩阵）相加减，等于标量与矩阵中的每个元素相加减。这一点请读者特别注意。

（2）矩阵的乘法

我们在命令区中输入：

A=magic(3)

B=pascal(3)

C=A*B

得到的结果如图 4-3 所示。

如果 A 的列数与 B 的行数不相等，那么这两个矩阵相乘会得到什么结果呢？我们来试一试。

在命令区中输入：

A = magic(3)

B = pascal(4)

C = A*B

得到图 4-4 所示的结果。

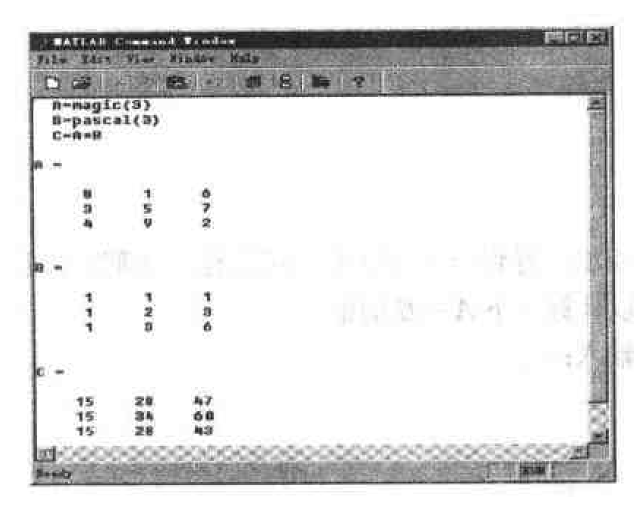


图 4-3　矩阵相乘

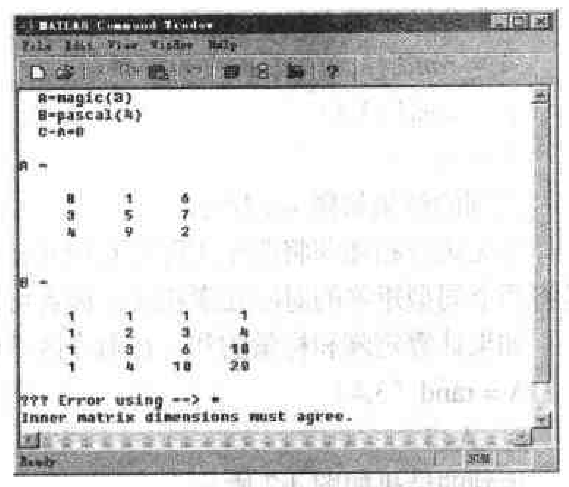


图 4-4　矩阵相乘出错信息

系统会自动告诉使用者，乘法无法进行，原因是两个矩阵列数和行数不匹配，无法进行矩阵乘法。只有修改合适后，才能继续进行计算。

再看一个例子，求 Kronecker 张量积。

输入：

A=[2,3,4;5,6,7]；

B=[1,2;3,4]；

C=kron(A,B)

得到下面的结果：

C =

2　4　3　6　4　8

6　8　9　12　12　16

5　10　6　12　7　14

15　20　18　24　21　28

（3）数组的简单计算

数组运算无论对于哪种运算操作都是对元素逐个进行的。MATLAB 设计这种运算的目的在于使大批数据的处理与标量情况相似。这样可以大大简化使用和编程。

下面是 MATLAB 系统中提供的数组运算符：

.＋ 加法　.－ 减法　.* 乘法　.^ 幂　.\ 左除　./ 右除　.′ 转置

读者应该注意到数组运算符比矩阵运算符多了一个小黑点。在数组计算过程中，小黑

点千万不能省略，否则将不按数组运算法则进行计算。参与数组计算的数组必须同维，或者其中的某个数组为标量。数组运算的结果仍是数组，且与原数组同维。

在命令区中输入：

```
A=[1,2,3,4,5];  B=[2,3,4,5,6];
C=A.*B
D=A.*2
```

得到结果如图 4-5 所示。

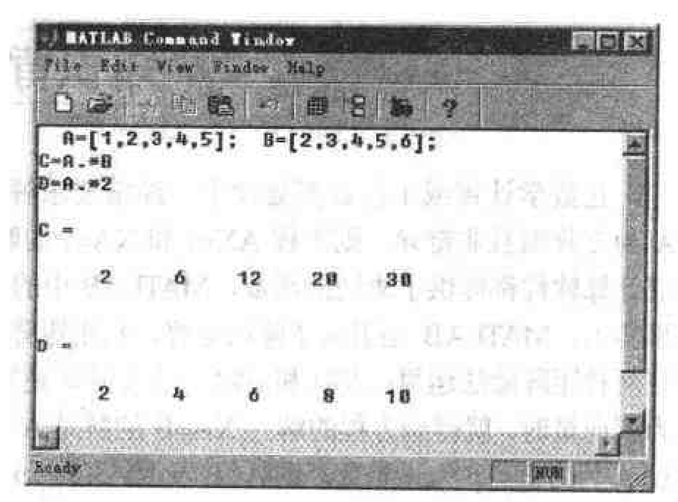


图 4-5　数组相乘

数组加减法比较简单，请读者自己练习。例如：在命令区中输入：

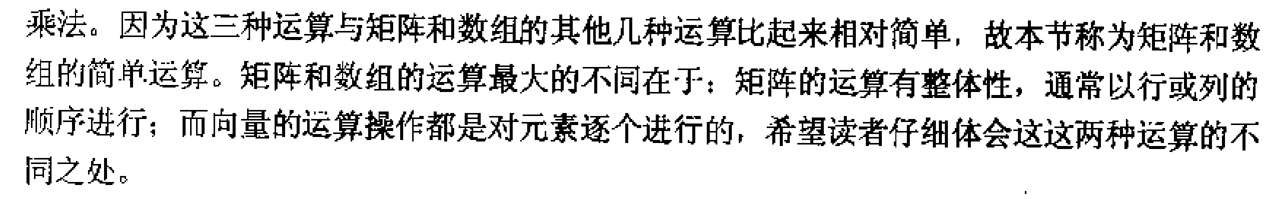
```
A=[1,2,3,4];
B=[4,3,2,1];
C=A+B
```

将得到：

```
C=
   5    5    5    5
```

【练习小结】

本练习主要向读者介绍了 MATLAB 中矩阵和数组的加减法和


乘法。因为这三种运算与矩阵和数组的其他几种运算比起来相对简单，故本节称为矩阵和数组的简单运算。矩阵和数组的运算最大的不同在于：矩阵的运算有整体性，通常以行或列的顺序进行；而向量的运算操作都是对元素逐个进行的，希望读者仔细体会这这两种运算的不同之处。

【思考题】

1．请创建一个 3×3 矩阵和 4×4 矩阵，判断能否相加减，能否相乘？

2．设 A 和 B 是两个同阶方阵判断能否相加减，能否相乘。

3．若 1 中的矩阵不能进行运算，试改变矩阵的维数，使其能够相加减或相乘。并计算结果。

4．A=rand(3)，B=magic(3)，C=rand(3,4)，计算：A*B*C

5．A=[1,2,3]计算 A′ 与 A 的积，A*B 与 B*A 是否相等？

练习 5　矩阵的除法与乘方运算

知识背景

在数学计算或工程数据处理中，经常要求解线性方程组，这就涉及到求逆矩阵。如果 A 为方阵而且非奇异，则方程 AX=I 和 XA=I 的解称为矩阵 A 的逆，用A^{-1}表示。一般的数值计算软件都提供了求逆的函数，MATLAB 中的求逆函数为 inv（A），其中 A 必须是方阵。事实上，MATLAB 还引入了除法运算，利用矩阵除法也可以求解线性方程组。MATLAB 中有两种矩阵除法运算：左除和右除。设 A 是可逆矩阵，AX=B 的解是 A 左除 B,即 X=A\B（B 为列向量时，就得到方程的解）；XA=B 的解是 A 右除 B，即 X=B/A。什么是矩阵的乘方呢？设 A 为方阵，p 为正整数，则 A^P 表示 A 自乘 p 次。若矩阵 A 为方阵且非奇异，A^（－P）表示矩阵 A^{-1} 自乘 p 次。

主要内容

【本练习考查知识点】

MATLAB 提供的矩阵除法运算有着丰富的内涵，而矩阵的乘方也是我们经常会遇到的运算。我们在这个练习中将学习矩阵的除法运算，弄清楚左除与右除的区别，并能利用除法求解简单的线性方程组，然后与由矩阵求逆法得到的结果加以比较。本练习还将学习矩阵的乘方，对于指数为正整数、负整数和分数的情况分别加以讨论。

练习过程

（1）求解逆矩阵

MATLAB 的函数 inv（A）可用来求 A 的逆矩阵，但当 A 为长方阵时，方程 AX=I 和 XA=I 至少有一个无解，无法求得矩阵的逆。但 MATLAB 还提供了 pinv（A）命令，求得的结果叫做矩阵 A 的伪逆。伪逆可以在一定程度上反映逆矩阵的特点。

我们在命令区输入：

```
A=[1,2;3,4;5,6];
B=pinv(A)
```

C=B*A

D=A*B

运行结果如图 5-1 所示。

从运行结果中我们看到，B*A（B 为 A 的伪逆矩阵）的结果 C 是 2x2 单位阵，而 A*B 所得的结果 D 不是单位阵。请读者加以区分。如果我们在命令区中输入：

A=[1,2;3,4;5,6];

B=inv(A)

则工作区中会显示如下信息：

??? Error using ==> inv

Matrix must be square.

系统提醒使用者矩阵不是方阵，可见 inv 函数只能对方阵求逆。但是，pinv 函数却能对方阵求逆。也就是说，pinv 包含了 inv 函数的功能。

（2）矩阵的除法

“\” 和 “/” 分别表示左除和右除。这两种运算符号的功能是不同的。

A\B 表示 A 左除 B；A/B 表示 B 右除 A；B\A 表示 B 左除 A；B/A 表示 A 右除 B。

我们通过具体的例子来看一下它们的区别。

我们在命令区中输入：

A=[1,2;3,4;5,6];

B=[5,6;7,8;9,10];

C=A\B

D=A/B

得到的结果如图 5-2 所示。

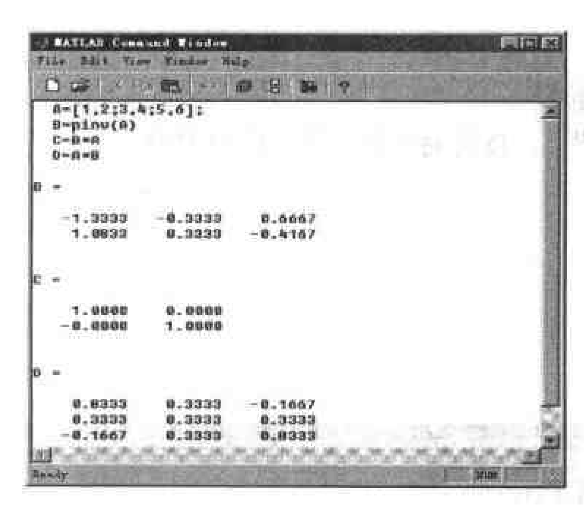


图 5-1　求解伪逆矩阵

图 5-2　左除与右除

同样在命令区中输入：

A=[1,2;3,4;5,6];

B=[5,6;7,8;9,10];

E=B\A

F=B/A

将得到如下结果：

```
E =                        F =
  5.0000    4.0000            0        0    1.0000
                          -0.5000      0    1.5000
 -4.0000   -3.0000        -1.0000      0    2.0000
```

（3）矩阵求逆和矩阵除法解方程组

对于 Ax=b 这样的线性方程组，我们有两种解法。一种是利用矩阵求逆，即 x=inv(A)*b；另一种方法是用左除，即 x=A\b。这两种解法的算法不同，用除法解方程的速度要快于矩阵求逆法。求逆函数用的是高斯消去法，除法直接使用高斯消去法，所以速度要快一些。通过下面的例子，我们可以看到这两种方法的精度基本相同，因此，应尽量选用除法。

我们在命令区中输入：

```
A=rand(5);
b=ones(5,1);
x1=inv(A)*b
X2=A\b
```

得到的结果如图 5-3 所示。

（4）矩阵的乘方运算

对于方阵的乘方运算 $A\hat{}p$，我们根据 p 的取值不同来分几种情况来讨论。

假设矩阵 A 为方阵且非奇异。则我们可以得到下面的一些计算规则。（请注意：MATLAB 有着很好的程序可读性，会根据用户输入数据自身的特点进行相应的计算。因此一定要注意输入的准确性，否则 MATLAB 会按错误的输入进行计算。）

① 当 p 为正整数时，$A\hat{}p$ 表示 A 自乘 p 次。

② 当 p 为负整数时，$A\hat{}(-p)$ 表示矩阵 A^{-1} 自乘 p 次。

③ 当 p 为 0 时，$A\hat{}0$ 等于与 A 同维的单位阵。

④ 当 p 分数时，若 A 可以分解为 $A=WDW^{-1}$，D 为对角阵，则 $A^p=WD^pW^{-1}$。

我们在命令区输入：

```
A=[1,2,3;2,3,1;3,2,1];
B=A^2
C=A^0.3
```

得到的结果如图 5-4 所示。

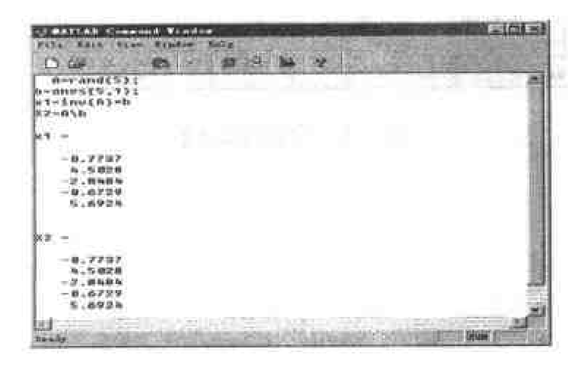


图 5-3　求逆法和除法解线性方程组

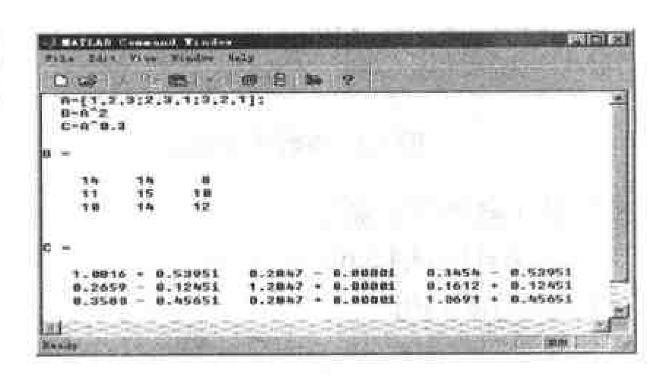


图 5-4　矩阵的乘方

请注意：如果 A 的特征值有重根，则指令不执行；有些矩阵的非整数次方可能存在多个解，MATLAB 只显示一个解。

我们也可以求标量的矩阵乘方 $p^{\wedge}A$

若 A 可以分解为 $A = WDW^{-1}$，D 为对角阵，则可以定义标量矩阵乘方为：

$$p^{\wedge}A = W\begin{bmatrix} p^{d_{11}} & & \\ & \Lambda & \\ & & p^{d_{nn}} \end{bmatrix}W^{-1}$$

取上例中的 A，计算 B=(2)^A，得到的结果如下：

```
B =
    21.2021    24.8000    17.9979
    20.3688    26.8000    16.8312
    20.9521    24.8000    18.2479
```

【练习小结】

在这个练习当中，我们学习了矩阵求逆和伪逆的方法，认识到 pinv 命令是比 inv 更强的命令。初步接触了矩阵的除法，了解了左除和右除的区别。并且通过求解简单线性方程组，比较了矩阵求逆和矩阵除法两种方法的特点。矩阵的乘方运算情况稍微复杂些，要分情况讨论。

【思考题】

1. 创建一个 5×5 随机矩阵，并求逆。
2. 创建一个 4×3 随机矩阵，求逆。用到的函数命令与第一题是否相同，能否互换？
3. 用两种方法求解 Ax=b 的解。（A 为 4 阶随机矩阵，b 为 4 阶列向量）
4. 调用上题中的 A，计算 A 的 5 次方。
5. 还是利用上面的 A，计算 $(0.5)^{\wedge}A$ 的值。
6. 说说第 4 题、第 5 题分别是什么运算？在 MATLAB 中各有什么含义？

练习 6　标量和向量函数及命令区操作

知识背景

MATLAB 为我们提供了大量的函数。按照用途，可以分为三类：标量函数、向量函数和矩阵函数。顾名思义，标量函数作用于标量，一般用于简单的数值计算，向量函数只作用于行或列向量。矩阵函数又可以分为两类：构造函数（用于创建）和计算函数（用于计算）。我们在本节介绍标量函数和向量函数，而矩阵函数作为 MATLAB 中的重点内容，我们在下个练习当中单独介绍。

MATLAB 命令区是用户用来输入命令、调用函数、计算结果的窗口。MATLAB 中关于命令区的命令也比较多。我们在前面的练习里学习了一部分命令区操作的内容，但远远不够，在本练习中我们来继续加深这部分内容的学习。

主要内容

【本练习考查知识点】

在本练习当中，我们学习 MATLAB 的基本标量函数和向量函数，求解简单的线性方程组，然后与由矩阵求逆法得到的结果加以比较。本练习还将学习矩阵的乘方，对于指数为正整数、负整数和分数的情况分别加以讨论。

练习过程

（1）标量函数

在我们遇到的计算中，很多是较为简单的标量计算。对于一个数学表达式，我们怎么将它转化为计算机能够识别的语句函数呢？从最初的机器语言到后来的汇编语言以及现在的高级语言，我们已经不用为这个问题而担心了。MATLAB 提供了大量的函数命令，可以帮助我们完成这些工作。我们只需要简单地输入数学表达式，就可以进行计算了。

MATLAB 中常用的标量函数如下：

三角函数：sin、cos、tan、cot、sec、csc、asin、atan、acot、asec、acsc、sinh、cosh、tanh、asinh、acosh、atanh。

其他标量函数：sqrt、exp、log、log10、abs（绝对值或复数模）、round（四舍五入取整）、floor（向 $-\infty$ 取整）、ceil（向 $+\infty$ 方向取整）、sign（符号函数）、real（取实部）、imag（取虚部）、angle（取幅角）、rats（有理逼近）。

上面的函数用于标量，如果用于矩阵或数组，会产生什么结果呢？我们来看一个例子。

在命令区中输入：

```
x=[1.3,2.4,3.7,4.3,5.8];
y=sin(x)
z=round(x)
w=floor(x)
```

得到的结果如图 6-1 所示。

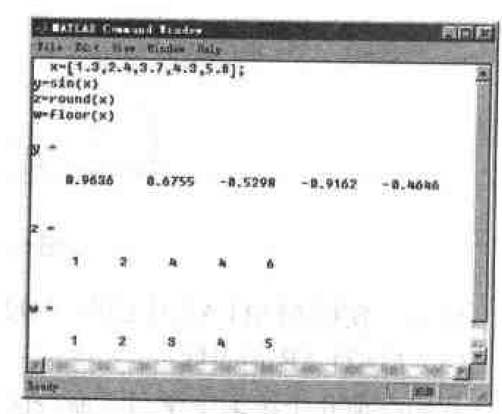


图 6-1　标量函数作用于矩阵

读者很容易看到，标量函数作用于矩阵（或数组）时，是作用于矩阵（或数组）的每一个元素。这个功能将大大方便我们处理成批的数据。

另外，MATLAB 还提供了一个计算函数的函数命令：fecal（F，x），其中 F 是函数名，x 是要处理得数据。例如，我们在命令区中输入：

```
x= [1.3,2.4,3.7,4.3,5.8];
z= fecal ('round', x)
```

得到的结果如下：

z =　1　　2　　4　　4　　6

与前面计算的 y 值比较，结果完全一样。

（2）向量函数

MATLAB 中有些函数只有当它们作用于行或列向量时才有意义，称为向量函数。当然，这些函数也可以作用于矩阵，这时它就会产生一个行向量，行向量的每个元素是函数作用于矩阵相应列向量的结果，常用的有：

Max（最大值）、min（最小值）、sum（和）、length（长度）、mean（平均值）、median（中值）、prod（乘积）、sort（从小到大排列）。

实际生活中，我们经常会遇到排序、求和的问题。比如对工程中得出的数据，需要迅速找出最大、最小和中间值，并要求计算所有数据之和这类问题，如果数据量较小，可以在普通计算器中完成。但要是面对一大堆数据，计算器就不能满足要求，使用上面的 MATLAB 函数，则可以迅速完成工作。

我们来看下一个例子。在命令区中输入：

```
x=[0.6833, 0.2126, 0.8392; 0.6288, 0.1338, 0.2071,
  0.6072, 0.6299, 0.3705, 0.5751];
  a=max(x)，b=min(x)，c=mean(x)，d=median(x)
```

得到的结果如图 6-2 所示。

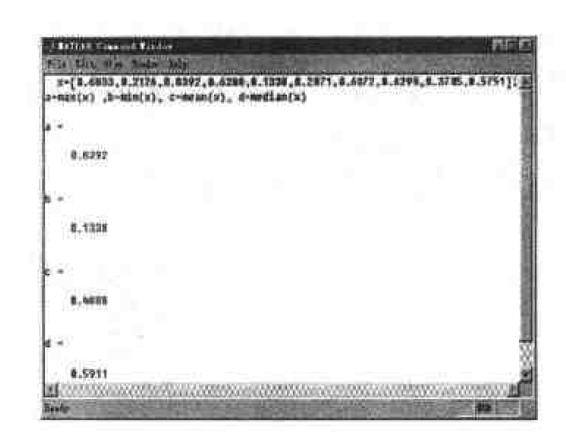


图 6-2　向量函数处理数据

可见，使用 MATLAB 中的向量处理函数能大大减少工作量。

（3）MATLAB 命令区

我们在前几个练习中学习了命令区菜单命令的“File”项和常用操作键，下面我们学习命令区中菜单栏的其他项和命令区中的工具栏。最后介绍一些通用操作命令。

① 菜单栏“Edit”选项

单击菜单栏上的“Edit”选项，将产生一个下拉菜单，如图 6-3 所示。

下拉菜单包含七项。“Undo”用于撤销上一次操作；“Cut”用来剪切所选内容；“Copy”用来复制；“Paste”用来粘贴前面剪切或复制的内容；“Clear”是清除命令，可用来清除尚未被执行的命令（即未按回车键）。当输入命令出现错误，想要删除时，可选中要删除的内容，然后单击“Clear”即可；“Select All”用于选中工作区里的所有内容；“Clear Session”用于清除显示的全部内容，但工作空间里的变量不被删除。

② 菜单栏“View”选项

菜单栏“View”选项用来设定是否在 MATLAB 主窗口中显示工具栏，并且可以在这里完成窗口切换。如果“Toolbar”前无勾号，则 MATLAB 主窗口不显示工具栏，这样可以让界面显得更紧凑些。

③ 菜单栏“Window”选项

单击菜单栏“Window”选项，就会显示已打开的 MATLAB 窗口的信息。

④ 菜单栏“Help”选项

单击菜单栏“Help”选项，将产生一个下拉菜单，如图 6-4 所示。

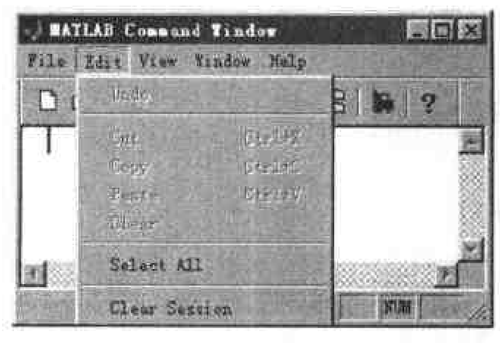


图 6-3　“Edit”下拉菜单

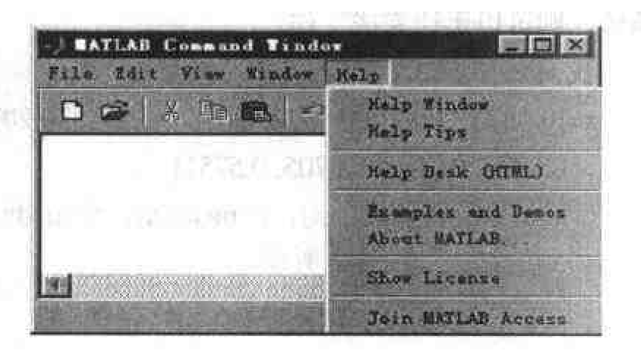


图 6-4　菜单栏“Help”选项

我们看到，“Help”下拉菜单一共包含 7 项。“Help Window”用于打开 MATLAB 的帮

助窗口；“Help Tips”也用来打开 MATLAB 的帮助窗口，但是会显示 MATLAB 帮助系统的分类和使用方法；“Help Desk（HTML）”用于打开以 web 页面显示的帮助信息，单击“Help Desk（HTML）”和我们以前介绍过的在工作区中直接输入“Helpdesk”的效果是一样的。

“Examples and Demos”用来演示 MATLAB 自带的演示例子，相当于在工作区中直接输入“demos”；“About MATLAB”、“Subscribe”、“Show license”和“Join MATLAB Access”用来显示与 MATLAB 软件和厂商相关的信息。

MATLAB 的工具栏由图标组成，它们的功能如表 6-1 所示。

表 6-1　工具栏功能介绍

图　标	功　能
	新建一个 M 文件，相当于菜单命令“File”中“New”的命令
	打开一个已有文件，相当于菜单命令“File”中的“Open”命令
	将选中的内容删除，相当于菜单命令“Edit”中的“Cut”命令
	复制选中的内容，相当于菜单命令“Edit”中的“Copy”命令
	将选中的内容粘贴到工作区，相当于菜单命令“Edit”中的“Paste”命令
	撤销上一步操作，相当于菜单命令“Edit”中的“Undo”命令
	工作区管理，相当于菜单命令“File”中的“Show Workspace”命令
	路径管理，相当于菜单命令“File”中的“Set Path”命令
	SIMULINK 类管理，用于建立新仿真类型
	显示帮助窗口

MATLAB 还提供了很多键盘控制指令，有时使用键盘操作，能够达到比通过菜单命令控制更为快捷的效果。下面是 MATLAB 命令窗口的一些通用操作指令。

表 6-2　MATLAB 通用键盘指令

指　令	功　能
cd	改变当前工作目录
clear	清除内存中的所有变量和函数
clc	清除 MATLAB 工作区中所有显示的内容
clf	清除 MATLAB 当前窗口中的图形
dir	列出指定目录下的文件和子目录清单
disp	在运行中显示变量或文字内容
echo	控制是否显示运行文字指令
hold	控制是否刷新当前窗口
pack	收集内存碎块以扩大内存空间
type	显示所指定文件的全部内容
quit	关闭 MATLAB

【练习小结】

本练习我们首先学习了标量函数和向量函数的概念，认识了很多标量函数和向量函数命令，有了这些命令，我们可以完成所有数组型的运算。本练习介绍的都是最为普通的命令，这些命令都是在 MATLAB 中极为常用的。希望读者能够熟记这些命令，这对进一步学习 MATLAB 是极有好处的。

MATLAB 命令窗口有着良好的界面，它支持语句控制，熟悉 MATLAB 窗口，能够根据使用要求灵活操作，是一个 MATLAB 高手所必须具备的本领。本节系统地介绍了 MATLAB 的命令窗口和常用控制语句，希望这些内容能够对读者有所帮助。

【思考题】

1. 计算π 、π/2 的正弦和余弦值。
2. 计算$\sin 75^{\circ}-\sqrt[3]{9-\sqrt{2}}$
3. 如果要求矩阵当中所有元素的正切值，是否要重新编写数组进行计算，有没有什么好办法？
4. 对于大量工程数据，如何迅速找出其中的最大值？最小值和中值？
5. 如何迅速清除命令区显示内容，如何迅速得到 MATLAB 的联机帮助？

练习 7 矩阵函数

知识背景

MATLAB 的全称是 Matrix Laboratory，意思是矩阵实验室。可见，MATLAB 最重要的功能在于矩阵运算。众所周知，随着科学技术的发展，线性问题广泛存在于自然科学和技术科学的各个领域，某些非线性问题也能够转化为线性问题来处理。而矩阵是线性代数中最基本的工具。所有线性代数问题归根到底都是矩阵问题。掌握了矩阵的计算方法，也就能够掌握代数方法和几何方法去处理科学技术中遇到的难题。在本练习当中我们就来学习 MATLAB 中重要的部分：矩阵函数。

主要内容

【本练习考查知识点】

本练习首先介绍 MATLAB 中常用的矩阵函数，包括求方矩阵的行列式，求一般矩阵的秩、矩阵的范数等。接着向大家介绍矩阵的分解 ，如何根据计算要求将原有矩阵分解，这也是计算中经常遇到的问题。在本练习中，你将看到有关这方面的内容。

练习过程

（1）首先我们来认识 MATLAB 中常用的矩阵函数。

我们将常用矩阵函数列表如下：

表 7-1　常用矩阵函数

函　　数	功　　能
cond（A）	求矩阵 A 的条件数
Det（A）	求方阵 A 的行列式
Dot（A，B）	矩阵 A 与 B 的点积
Eig（A）	方阵 A 的特征值和特征向量
Norm（A，1）	矩阵 A 的 1－范数
Norm（A）、norm（A，2）	矩阵 A 的 2－范数

（续表）

函　　数	功　　能
Funm（A，'fun'）	一般的方阵函数
Norm（A，'inf'）	矩阵 A 的无穷大范数
Norm（A，'fro'）	矩阵 A 的 F一范数
Rank（A）	矩阵 A 的秩
Rcond（A）	矩阵 A 的倒条件数
Svd（A）	矩阵 A 的奇异值分解
Trace（A）	矩阵 A 的迹
Expm（A）	求 e^A 的值
expm1（A）	用 Pade 法求 e^A 的值
expm2（A）	用求用 Tayor 级数求 e^A 的值
Expm3（A）	用特征值和特征向量求 e^A 的值（独立特征向量等于矩阵秩）
Logm（A）	求矩阵 A 的对数
Sqrtm（A）	求矩阵的平方根

我们在下面的例子中来看一下矩阵函数的用法：

我们在命令区中输入：

```
A=rand(4);
a=det(A)
b=trace(A)
c=rank(A)
d=norm(A,'inf')
e=norm(A,'fro')
```

得到如下的结果：

```
a =
    0.1155          （求得 A 的行列式值）
b =
    2.7334          （求得 A 的迹）
c =
    4               （求得方阵 A 的秩）
d =
    3.5846          （求得方阵 A 的无穷大范数）
e =
   2.5716           （求得方阵 A 的 F 范数）
```

从上面的例子可以看出，有了这些函数，我们将能够很方便、快捷地计算出矩阵的相关数值。

那么，矩阵函数和我们在上个练习中学到的标量函数和向量函我有什么区别呢？

如果我们在命令区中输入：

```
A=[1/6,1/2;2/3,5/6]*pi;
B=sin(A)
```

C=funm(B,'sin')

得到的结果如图 7-1 所示：

从图中我们可以清楚地看到，对于一个矩阵，数组型函数（标量函数和向量函数）是对矩阵中的每一个元素进行运算，而矩阵函数则先进行矩阵特征值分解，然后再进行计算。这是很重要的区别，请读者注意。

（2）矩阵分解

在代数运算中，往往要牵涉到矩阵分解。它是矩阵计算的基础，是很重要的一环。下面我们来学习分解矩阵。

我们来学习解线性方程组经常要用到的 LU 分解。LU 分解，又称作高斯消去法，它可以把任意方阵分解成下三角矩阵的基本变换形式和上三角矩阵的乘积。它的表达式为：A＝LU。其中，L 为下三角矩阵的基本变换形式，U 为上三角矩阵。

比如说，我们有矩阵 A＝[2,3,4;6,7,8;1,5,9]，想要分解成两个矩阵的乘积。

首先在命令区中输入：

A＝[2,3,4;6,7,8;1,5,9]

[l,u]=lu(A)

按回车后，将得到如图 7-2 所示的结果。

从结果中我们可以清楚地看到，矩阵 A 被分解成了两个矩阵的积，其中，u 是一个下三角阵。

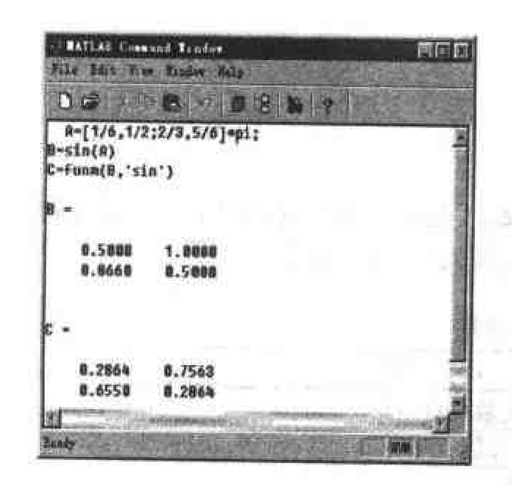


图 7-1　矩阵函数与数组型函数比较

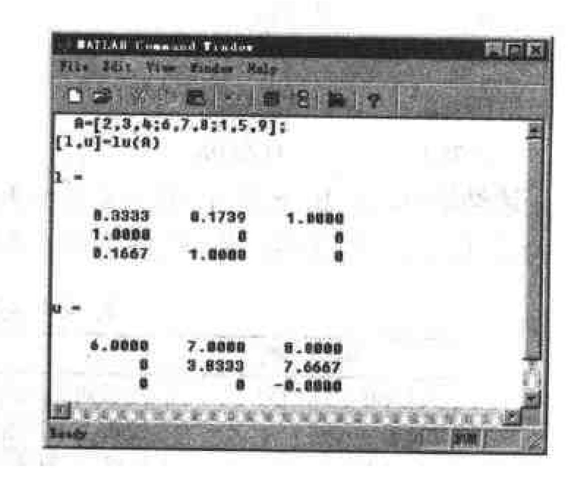


图 7-2　矩阵 LU 分解

矩阵 LU 分解是一个非常有用的分解函数，MATLAB 里矩阵的求逆和求行列式都是通过 LU 分解来实现的。

我们再来看一下如何进行奇异值分解。

不妨设矩阵 A 的行数大于列数，b 是矩阵 A 中 n 个奇异值所构成的列向量。若 A=[1,2;3,4;5,6]，如何对它进行奇异值分解呢？我们来使用 svd 函数命令。

在命令区里输入：

A=[1,2;3,4;5,6];

[a,b,c]=svd(A)

[e,f,g]=svd(A,0)

得到以下结果：

```
a =
    0.2298    0.8835    0.4082
    0.5247    0.2408   -0.8165
    0.8196   -0.4019    0.4082
b =
    9.5255         0
    0         0.5143
    0              0
c =
    0.6196   -0.7849
    0.7849    0.6196
e =
    0.2298     0.8835
    0.5247     0.2408
    0.8196    -0.4019
f =
    9.5255         0
    0         0.5143
g =
    0.6196     -0.7849
    0.7849      0.6196
```

上述结果中，a、b、c 是奇异值分解的结果；d、e、f 是以简单形式给出的奇异值分解结果。为了让读者一目了然，我们将矩阵的分解命令总结如表 7-2 所示。

表 7-2　矩阵分解函数

函　数	功　能
cdf2rdf（V，D）	复数对角形转换成实数块对角形
cchol（A）	矩阵 A 的 Cholesky 分解
eig（A）	矩阵 A 的特征值分解
Hess（A）	矩阵 A 的 Hessenberg 形式
LU（A）	矩阵 A 的 LU 分解
null（A）	由奇异阵分解得出的矩阵 A 的零空间的标准正交基
orth（A）	矩阵 A 行向量的标准正交基
pinv（A）	求矩阵 A 的伪逆
qr（A）	矩阵 A 的 QR 正交三角形分解
qz（A）	矩阵 A 的 QZ 分解，用于广义特征值
rref（A）	将矩阵 A 转化为逐行递减的阶梯阵
rsf2csf（V，D）	实数块对角形转化成复数对角形
schur（A）	矩阵 A 的 Schur 分解
subspace（A）	计算由 A、B 张成的子空间的夹角
svd（A）	方阵 A 的奇异值分解

请读者对照上表，练习几个命令。相信读者能够举一反三，最终熟练使用矩阵分解命令，从而完成矩阵计算。

【练习小结】

本练习主要向大家介绍了 MATLAB 中常用的矩阵函数和分解函数。如何利用矩阵函数进行矩阵运算，如何利用矩阵分解函数恰当地将原有矩阵分解为新的矩阵，这些内容都要求读者掌握。这里特别强调指出的是 LU 函数。因为它是与解线性方程组有关的一个非常重要的函数，所以请读者熟练掌握。

【思考题】

1. 创建一个矩阵，然后求出它的秩。这要用到什么命令？
2. MATLAB 中有几种求范数的函数命令，它们的区别是什么？
3. 矩阵函数与数组型函数有什么区别？
4. 矩阵分解除了 LU 命令外，还有那几种？分解后的矩阵是什么形似的矩阵？
5. 试分解第 1 题中创建的矩阵。

练习 8　多项式的表达和一元方程求根

知识背景

多项式是我们在初中就接触到的数学概念，频繁出现在很多的计算里，我们所学过的很多公式、定理都是以多项式形式表述的。在 MATLAB 中，属于多项式的天地也很广阔，比如求根、分解、求导数以及多项式的拟合，这些功能在 MATLAB 中都能得到很好的实现。

解一元方程是平时学习和工程应用中最常遇到的问题。一元一次方程方程是非常容易都出来的，一元二次方程可以用求根公式来都，但是如果碰到三次和三次以上的一元方程就非常麻烦了，一般没有解析根，只能近似求都。我们可以采用高级语言都程，用插值法或者逼近法求都。MATLAB 提供了非常都单的方法，只需要用一个 fzero 函数，就可以方便地求出复杂的高次方程根。

主要内容

【本练习考查知识点】

本练习先向读者介绍多项式的一般表达方式，包括多项式系数、矩阵的特征多项式等。然后介绍数定义的方法，以及如何利用 fzero 函数对一元方程求根。需要掌握 fzero 函数的调用格式，另外还介绍了求一元函数极值的函数命令 fmin。

练习过程

（1）首先我们来学习多项式的表达。

多项式在 MATLAB 中是由行向量来表达的，向量中的元素是多项式系数的降幂排列。其中最后一个元素代表多项式中的 0 幂项，即常数项，这一项千万不可省略，否则 MATLAB 将无法识别这一项。

比如说我们都要输入一个四次多项式，在 MATLAB 中如何实现呢？其实很简单，只需要在工作区中输入系数矩阵就可以了。请看下面的例子。

想要都入 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$，只需要在命令区中输入：

p=[0.69552，0.436，0.668，1.35]

这样就建立了 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 这个多项式。

现在我们来求这个多项式的根。在命令区中输入：

```
p=[0.69552，0.436，0.668，1.35]
x=roots(p)
```

回车后，将得到如图 8-1 所示的结果。

上面的求根方法是先把多项式转化为伴随矩阵，然后再求特征值，可靠性和精度都高于经典方法。因为采用矩阵算法，所以 MATLAB 中的多项式和它的根都是向量。如图 8-1 所示，多项式为行向量，根为列向量。

那么，如果把问题反过来，知道了多项式的根，能否求出多项式的系数行向量呢？我们接着上面的例子，在工作区中输入：

```
p=[0.69552,0.436,0.668,1.35];
x=roots(p);
pp=poly(x)
```

回车后，将得一个多项式的系数行向量，但这个行向量和 p 的形式并不一致，于是我们将得到的结果乘以原多项式最高次项的系数，形式就变得完全一样了。请注意，MATLAB 中，由根构造多项式时，结果中行向量的第一项系数为 1。

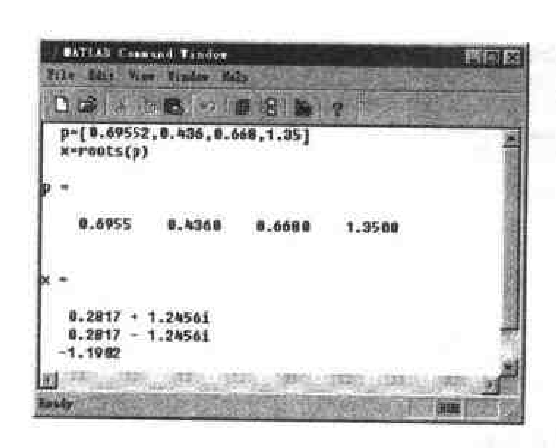


图 8-1　多项式求根　　　　图 8-2　由根构造多项式

（2）我们再来看一下对于一个多项式，如果不用 roots 命令如何求根。首先我们来学习函数的定义。MATLAB 的数学函数表示是在 M 文件中进行的（有关 M 文件的内容后面还要详细介绍）。打开菜单“File”>“New”>“M-file”，新建了一个编辑窗口，如图 8-3 所示：

要想输入以下的函数表达式：

$$f(x)=1.35+0.668x+0.758x^2-0.322x^2(1-2.16x)$$

我们在编辑窗口中输入：

```
function y=demax(x);%定义函数demax(x)
y=1.35+0.668.*x+0.758.*x.^2－0.322*(1－2.16.*x).*x.^2;
```

然后单击“保存”按钮，弹出保存对话框，将路径设置为默认目录，此处是

C:\MATLABr11\work，将文件命名为 demax.m，如图 8-4 所示保存文件。

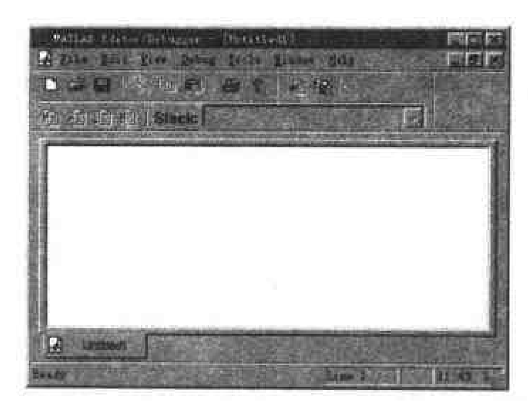


图 8-3　编辑窗口

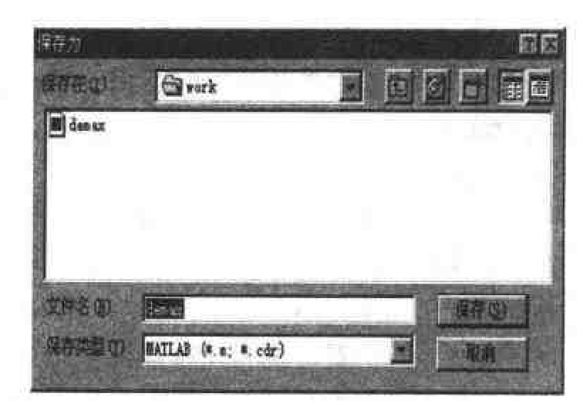


图 8-4　保存文件

（3）使用函数 fzero 可以找出函数值为零的点，寻找零点时可以指定一个开始点，或者指定一个区间。如果指定一个开始点，此函数首先在开始点附近寻找一个使函数值变号的区间。如果我们知道函数值在某个区间上变号，则可以用一个包含两个元素的矢量指定区间作为 fzero 的输入参数。现在回到 MATLAB 的命令窗口，在命令窗口中输入：

a=fzero('demax',[−10,10])

得到输出结果如图 8-5 所示。

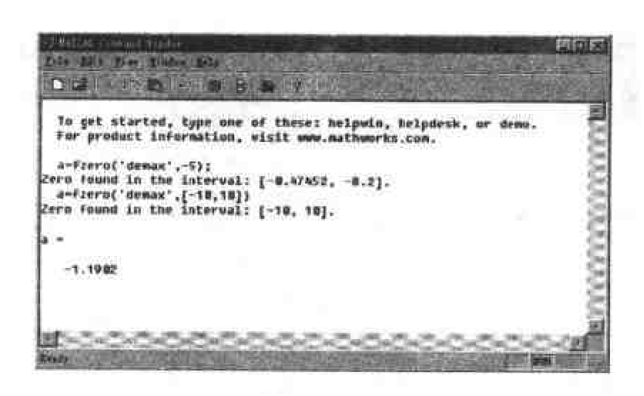


图 8-5　输出结果

求出的一元方程的根是−1.1902。

从上面的解法我们看出，使用 M 文件和求 0 值的方法显得数学性更强一点，但与直接使用 “roots” 命令相比，又显得很麻烦。关于 M 文件，我们在后面的练习中还要详细叙述，

本练习只是让读者对 M 文件有一个感性认识，事实上，在后面的练习中，读者就会发现 M 文件在复杂计算中是非常有用的。

（4）MATLAB 还可以用函数 fmin 求得指定区间上的局部最小值（以前版本的是 fminbnd），例如求函数

$$f(x)=x^2-2x-1+\frac{1}{(x-0.8)^2}$$

在开区间（−2，0.5）的最小点，仿照上面的步骤将函数命名为 ken.m。

在命令窗口中输入：

a=fmin('ken',−2,0.5)

得到结果为：

```
a=fmin('ken',-2,0.5)
a =
    -0.1535]
```

【练习小结】

本练习我们学习了多项式的表达。多项式在 MATLAB 中是由行向量来表达的，它的根是由列向量来表达的。我们可以有多项式求根，也可以由根构造多项式。

我们还利用 fzero 函数求一元方程的根、用 fmin 求一元函数的极小值，需要熟练掌握 MATLAB 中函数的表示方法，以及这两种函数的调用格式和相应参数的意义。另外，还要建立起对 M 函数文件的感性认识。

【思考题】

1．请在 MATLAB 中输入一个一元函数并保存，例如：

$$f(x)=x^3+(x-0.98)^2/(x+1.25)^3-5(x+\frac{1}{x})$$

2．用两种方法求上述函数的根。

3．求上述函数的极值。

4．如果函数不保存在默认目录下会有什么后果？

5．构造一个多项式，求根，再由根求出原来的多项式。得到的多项式与原来的多项式是否相同，为什么？

6．MATLAB 求多项式的根是用什么方法，与传统方法相比有什么优点？

练习 9　多项式的计算

知识背景

多项式的形式一般为 $f(x)=a_n x^n + a_{n-1}x^{n-1} + \Lambda\ a_1 x + a_0$ 。多项式的形式很简单，一般都是按未知量的降幂排列的各项之和。Matlab 中有关多项式的命令与矩阵相比较而言也显得要简单一些。对于一般多项式的运算，即加、减、乘、除、求导和估值等，Matlab 基本上都提供了相应的函数命令。利用这些函数命令，我们能够较松完成多项式的计算，从而大大降低实际工作的工作量。

主要内容

【本练习考查知识点】

本练习我们介绍关于多项式的常见计算，包括多项式的加减法，多项式的乘除法和求导、求函数值运算。在多项式加减法中，我们将回顾向量加减运算。多项式乘除法要用到向量卷积的概念。求导与求函数值也是需要理解的内容。

练习过程

（1）多项式的加减法

如果我们要求两个同阶多项式的和，只需要按照数组加法，将代表多项式的行向量相加就可以了。例如：

求 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 与 $f(x)=2.3+0.453x+2.342x^3$ 之和。

我们在命令区中较入：

```
p1=[0.69552,0.436,0.668,1.35]，p2=[2.342,0,0.453,2.3]
p=p1+p2
```

得到的结果如下：

```
p =
```

3.0375 0.4360 1.1210 3.6500

即结果为 $f(x)=3.6500+1.1210x+0.4360x^2+3.0375x^3$

我们来看一个例子：

求 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 与 $f(x)=1.23+0.645x+0.876x^2$ 之和。

我们在命令区中输入：

```
p1=[0.69552,0.436,0.668,1.35]
p2=[0,0.876,0.645,1.23]
p=p1+p2
```

得到的结果如下：

```
p =
0.6955    1.3120    1.3130    2.5800
```

与第一个例子相比，这是两个不同次多项式相加。其实也很简单，只要在低次多项式行向量相应处补上 0 元素，代表没有相应次数项即可。

（2）多项式的乘法

对 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 与 $f(x)=2.3+0.453x+2.342x^3$ 如果我们要求它们的积，应该怎么实现呢？答案是使用卷积函数。那么什么是卷积呢？

对 m 维向量和 n 维向量来说，它们的卷积就是下面的形式：

$$c(j)=\sum_{i=1}^{j}a(i)b(j+1-i)$$

读者可以看到，c 向量的维数为 $m+n-1$，这一条可以看作对结果正确性的初步判断。

求上面两个多项式的积，在命令区中输入：

```
c1=[0.69552,0.436,0.668,1.35]
c2=[2.342,0,0.453,2.3]
c=conv（c1,c2）
```

得到的结果如图 9-1 所示。从图中我们看到，上述两个多项式的积为 $f(x)=3.1050+2.1479x+1.3054x^2+4.9589x^3+1.8795x^4+1.0211x^5+1.6289x^6$。

显然，新的多项式的次数为 6，正好是 4+4－1=7。

另外有一点读者应该知道，如果需要求好几个多项式的积，只需要多次使用 conv 命令即可。

（3）多项式的除法

多项式的除法是乘法的逆运算，我们用向量解卷积函数 deconv 来完成。

求 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 与 $f(x)=1.23+0.645x+0.876x^2$ 之商。

观察上面两个多项式，由数学知识可以知道，商应该是由一个多项式和余数两部分组成的。那么最后的商应该写成两部分：a 与 b。其中，a 是商多项式；b 是余数（当然也可以是多项式，只是次数小于除式）。我们用 deconv 函数命令作用于代表两个多项式的行向量系数

表达式。

在命令区里输入：

```
p1=[0.69552,0.436,0.668,1.35];
p2=[0,0.876,0.645,1.23];
[a,b]=deconv(p1,p2)
```

得到的结果如图 9-2 所示。

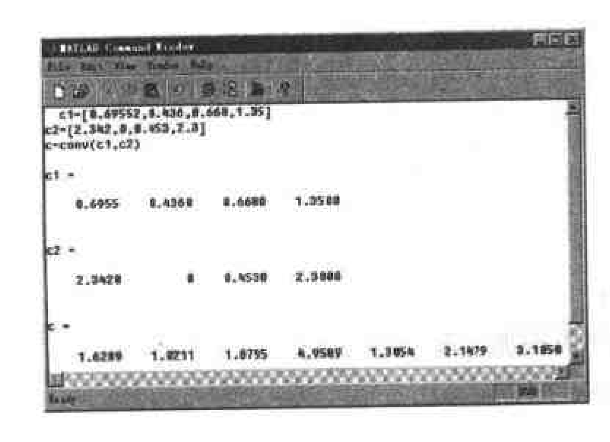


图 9-1　多项式乘积



图 9-2　多项式除法计算

最后的结果写成数学表达式是：

$$(1.35+0.668x+0.436x^2+0.69552x^3)/(1.23+0.645x+0.876x^2)$$

商式：$0.7940x-0.0869$

余式：$-0.2525x+1.4569$

（4）求多项式的导数

求多项式导数与进行多项式乘除法一样，属于比较简单的计算。只要使用 polyder 这个函数命令就可快速完成。

比如，我们想对 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 求导。

在命令区中输入：

```
p= [0.69552,0.436,0.668,1.35];
d=polyder(p)
```

得到如下的结果：

```
d =
  2.0866    0.8720    0.6680
```

如果想对一个分式求导，应该怎么实现呢？这时我们还是可以使用 polyder 函数。

求 $(1.35+0.668x+0.436x^2+0.69552x^3)/(1.23+0.645x+0.876x^2)$ 的导数。

我们在命令区中输入：

```
p1=[0.69552,0.436,0.668,1.35];
p2=[0,0.876,0.645,1.23];
[a,b]=polyder(p1,p2)
```

得到如下结果：

```
a =
    0.6093    0.8972    2.2625   -1.2926   -0.0491
    0.7674    1.1300    2.5710    1.5867    1.5129
b =
```

这个结果表示，所给分式的导数等于由 a 和 b 代表的多项式所组成的分式。其中 a 为分子，b 为分母。这样表示的好处在于分子与分母的形式类似，便于进行相关运算。读者可以自己验证一下这个结果。

（5）多项式值的计算

求多项式在某点的值是经常遇到的计算问题。对于这类问题，Matlab 的函数命令也能够非常迅速地解决。

例如，求 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 在 x=5.4 处的值。

我们利用 polyval 函数命令。

在命令区中输入：

```
p=[0.69552,0.436,0.668,1.35];
pa=polyval（p,1）
```

得到结果：

```
pa =
     3.1495
```

如果我们把问题扩展一下，要求多项式在很多给定点处的值，应该如何实现呢？显然，我们可以将单个点换成数组，然后进行计算。

例如，对于上面的多项式，我们要求 x 在 0、0.876、0.645 和 1.230 处的值.。

应该在命令区中输入：

```
p=[0.69552,0.436,0.668,1.35];
a=[0,0.876,0.645,1.230];
pa=polyval（p,a）
```

得到的结果如图 9-3 所示。

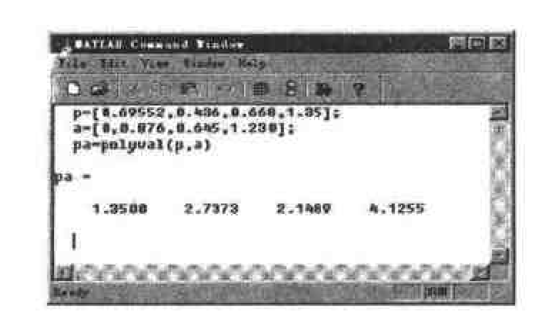


图 9-3 求多项式在多点的值

可见，求多项式在多点的值，只要按照上面的方法输入数据向量即可完成。

另外，polyvalm 函数按照矩阵运算规则计算多项式的值，这一点与 polyval 函数不同。

如果要对矩阵进行操作，可利用 polyvalm 函数仿照上面的方法进行计算。

【练习小结】

本练习向大家介绍了多项式的计算。多项式的四则运算遵循数组型运算法则，与矩阵运算不同。多项式求导和求值作为工程计算中经常遇到的问题，应该予以高度重视。总之，关于多项式的内容读者应该认真总结，争取达到熟练的程度。

【思考题】

1. 多项式的加减法在 Matlab 中实现的实质是什么？

2. 能否对多项式一次性多点求导？请对本练习中出现的相关多项式进行多点求导。

3. 请创建两个多项式，进行除法运算，然后交换位置，再进行除法运算。

4. 建立一个 5×5 矩阵。分别用 polyval 函数和 polyvalm 函数将矩阵代入 $f(x)=1.35+0.668x+0.436x^2+0.69552x^3$ 计算结果，进行比较。

练习 10　数值分析初步

知识背景

在科学研究和工程计算中，我们经常要求解诸如微分、积分这样一些问题。但是很多时候，我们无法得到它们的准确解，只能作一些近似计算。比如说$\int\frac{\sin x}{x}dx$这样的积分，形式虽然简单，但它的原函数无法由基本初等函数求得，所以我们只能用数值分析的方法来解这类问题。这样的例子还有很多，我们在以后的练习中还要接触。

主要内容

【本练习考查知识点】

本练习中，我们学习了数值分析的基本内容。主要介绍了数值微分，积分，求零点，求最小值这些经常遇到的计算。本节接触的是比较简单的内容。像数值积分的几种方法，只是让读者有一个感性认识，有关这部分较难的内容，我们在后面的练习中将陆续介绍。

练习过程

（1）简单数值积分

在直角坐标系里，求一个函数与 x 轴围成的面积，这类问题就属于简单积分。

比如我们求$\int_0^{\frac{\pi}{2}}\sin\ xdx$，我们可以用以下几种方法来计算。

① 将（0，π/2）10 等分，步长为π/20，按梯形算法计算。我们利用 trapz（x）函数进行积分。trapz（x）的功能是输入数组 x，输出为按梯形公式计算的 x 的积分。这是 MATLAB 中常用的数值积分方法。

在命令区中输入：

```
x=0:m:pi/2;
y=sin(x);
```

z=trapz(y)*m

得到下面的结果：

z =

　　0.9979

② 辛普森公式法：我们使用 quad（'fun'，a，b）函数命令。它将计算以 fun.m 命名的函数在区间（a，b）上的积分，自动选择步长。相对误差为 10^{-3}。

在命令区中输入：

z=quad('sin',0,pi/2)

得到的结果如图 10-1 所示。

③ 蒙特卡罗均值估计法：蒙特卡罗法是一种随机实验方法。关于蒙特卡罗均值估计法，在数值分析的书籍中一般都有介绍。本例使用 sum(x)函数，sum(x)函数的功能是输入数组 x，输出 x 的和。

我们在命令区中输入：

```
r=100000;
x=rand(1,r);
y=sin(x.*pi/2);
z=sum(y)*pi/2/r
```

得到的结果为：

z =

　　1.0010

上面三种方法得出的结果不同，这是由于所选方法各自产生的误差不同造成的。

（2）数值微分

数值微分与积分相比是比较困难的。本练习暂不作介绍，在后面的练习中，我们将作深入的学习。

（3）求极小值

练习 8 中我们学习了用 fmin 命令求函数最小值。现在，我们来学习另一个求最小值的命令：fminsearch。

在命令区中输入：

y='sin(x)'

z=fminsearch(y,－1,6)

得到的结果如图 10-2 所示。

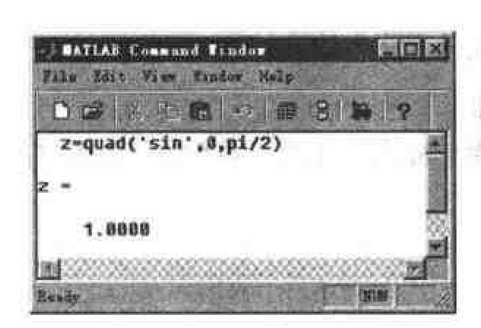


图 10-1　辛普森公式法求积分

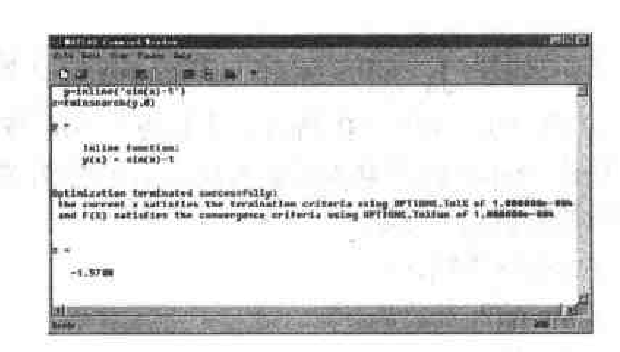


图 10-2　fminsearch 函数求最小值点

fminsearch 函数利用单纯形法求最小值。一般的形式是：

x=fminsearch(fun,x0,options,p1,p2,…)

其中，x0 是起始搜索点。

（4）求零点

练习 8 中，我们学习了 fzero 函数求零点的方法，这里我们再来看一个例子。求函数 humps 的零点

在命令区中输入：

```
xzero=fzero('humps',1)
  yzero=humps(xzero)
```

得到的结果如图 10-3 所示。

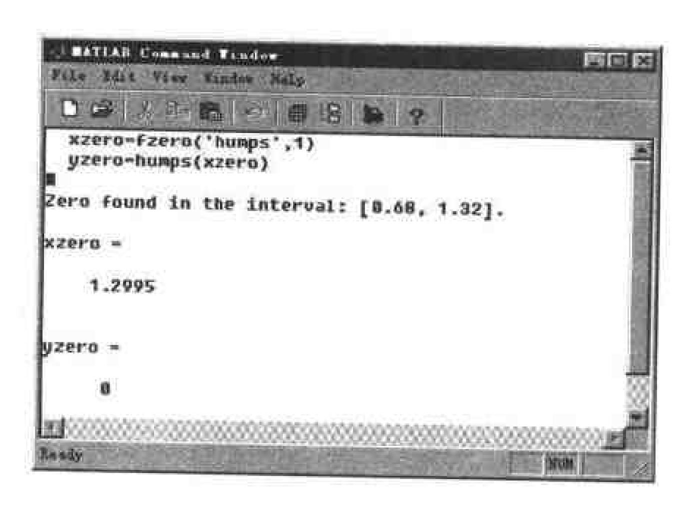


图 10-3 fzero 函数求零点

事实上，fzero 函数还可以寻找函数值等于任意值的点。比如要求$2x^2+1$等于 8 的点，我们可以定义一个新函数 f（x），使 f（x）=$(2x^2+1)$－8，然后求 f（x）的零点即可。两看是等价的。读者可以自己编制并在命令区里输入命令，然后通过运行结果验证一下我们的结论。

【练习小结】

本练习我们学习了以下内容：数值积分经常遇到的三种方法；求函数最小值常用的函数命令 fminsearch 的用法；求函数零点及任意值点命令 fzero 的用法，并且复习了函数定义的方法。这些内容十分基础，涉及的方法极为常用，希望读者好好掌握。

【思考题】

1．你所知道的数值积分方法有那几种，它们在 MATLAB 中是通过什么函数命令如何实现的？各有什么优缺点？

2．本练习介绍了求最小值的方法，那么，怎样利用 fmin 命令求最大值呢？你有什么好办法？

3．fminsearch 函数命令的特点是什么？请练习一下利用这个函数命令求指定函数最小

值，请总结使用时应该注意什么？

4．求函数$f(x)=3x^5+4x-9$等于 95 时的 x 值。

5．求零点在数学上有什么应用？什么情况下应该用这一方法？

6．利用 MATLAB 函数求出$f(x)=4x^6-x+x^3-93$在（0，100）上的最大值。

练习 11　关系和逻辑运算

知识背景

过去，当我们提到编程时，总是会想到 Fortran 或者 C，但应付它们繁琐的语句和冗长的代码始终是一件让人头痛的事。而 MATLAB 为我们提供了简洁、直观，更加符合人们平常思维习惯的代码，这一点使它比 Fortran 和 C 更加友好。MATLAB 中也包含了我们熟悉的关系运算符、逻辑运算符和语句循环、嵌套功能。这使得 MATLAB 不但有着强大的数值计算功能，而且成为良好的程序开发环境。本练习我们来熟悉最为基本的关系和逻辑运算。

主要内容

【本练习考查知识点】

本练习首先介绍 MATLAB 中的关系运算，介绍它们的表达形式和用法。接下来介绍逻辑运算，给出常用运算符和一些函数命令。有了上面的知识，我们可以进行关系和逻辑混合运算。本练习还复习了如何创建满足一定条件矩阵的方法。

练习过程

关系和逻辑运算：

关系运算符用来完成关系运算。在接制程序流程方面有着值为重要的作用。MATLAB 常用的关系运算符有：

<小于；> 大于；<= 小于或等于；>= 大于或等于；== 等于；~= 不等于 。

关系运算符可以用来比较两个数值，如果所描述的关系成立，则结果为 1，这表示逻辑真；反之，若所接述的关系不成立，则结果为 0，这表示逻辑假。

比如我们在命令区中输入：

a=1>2

b=1<2

得到的结果如图 11-1 所示。

由上面的结果看到，从简单的关系判断可以得到逻辑真或值。这在程序设计中是十分

有用的。

事实上，对于两个同型矩阵或同维数组，也可以用关系运算符来操作。利用这一点我们可以生成符合一定条件的新矩阵或数组。关于这一点，我们在前面的练习里已经有所介绍。在这里不妨再复习一下。

首先创建一个 5×5 矩阵 A（实际上是一个随机矩阵）如下，MATLAB 会保存这个变量。然后我们对 A 进行关系运算。

```
A =
    0.9501    0.7621    0.6154    0.4057    0.0579
    0.2311    0.4565    0.7919    0.9355    0.3529
    0.6068    0.0185    0.9218    0.9169    0.8132
    0.4860    0.8214    0.7382    0.4103    0.0099
    0.8913    0.4447    0.1763    0.8936    0.1389
```

在命令区内输入：

```
B=A>0.5
C=A~=0.5
```

回车后，将得到下面的结果：

```
B =
     1     1     1     0     0
     0     0     1     1     0
     1     0     1     1     1
     0     1     1     0     0
     1     0     0     1     0
C =
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
```

在这个运算中，MATLAB 将 A 中的每个元素按照给定的关系与 0.5 进行比较，关系成立，则相应元素位置为 1，不成立，则为 0。结果仍是 5×5 矩阵。

我们再来学习逻辑运算符。MATLAB 中的逻辑运算符有：

& 与；| 或；～ 非 。

具体意义如下表所示。

逻辑运算法则

Inputs		and	or	xor	not
A	B	A&B	A|B	xor（A，B）	~A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

根据表 11-1 的运算法则，我们可以很快得到下面两个逻辑运算的结果。

1|0 & 0 = 0

0 & 0|1 = 1

同样的，逻辑运算也可以应用于矩阵和数组，还可以与关系运算符配合起来使用。

我们在命令区中输入：

```
a=1:5
b=2:6
c=(a>2)&(b<5)
```

得到的结果如图 11-2 所示。

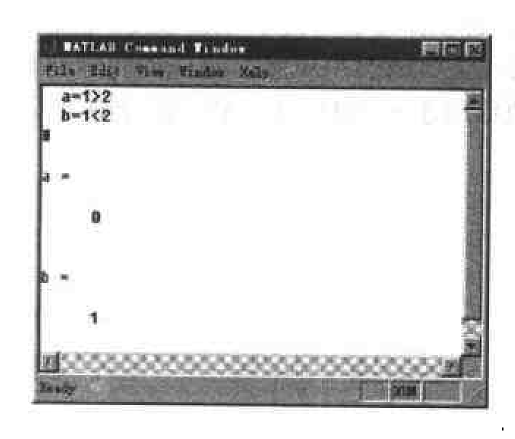


图 11-1　关系运算结果

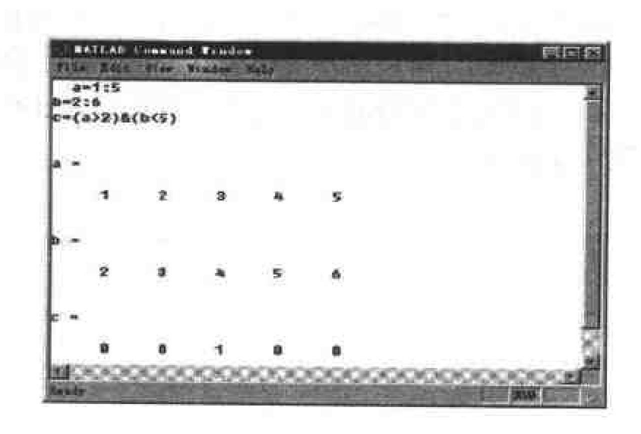


图 11-2　逻辑和关系运算

MATLAB 还提供了一些关系和逻辑函数。比较常用的有 all 和 any。它们一般的应用形式是：

y = all(x)：当 x 为数组时，只有 x 里的所有元素均不等于 0 时，y 为 1，其余情况为 0；x 为矩阵，则函数命令只作用于列元素，即对同列元素进行判断。

y = any(x)：当 x 为数组时，若 x 里有一个元素不等于 0，y 为 1，否则 y 为 0；x 为矩阵，函数命令也只作用于列元素。

比如我们在命令区中输入：

```
a=[1,2,0,4;2,0,1,3 ];
b=all(a) ,    c=any(a) ,    d=all(b) ,    e=any(b)
```

得到下面的结果：

```
b =
      1     0     0     1
c =
      1     1     1     1
d =
      0
e =
      1
```

另外，MATLAB 还提供了像 logical、xor 和 find 这样的逻辑函数，关于它们的用法，

读者可以自己通过联机帮助进行学习。

【练习小结】

本练习向大家介绍了 MATLAB 中的关系运算和逻辑运算。关系运算作用于数值、同型矩阵或同维数组上，结果是 0 或 1；逻辑运算有与、非、或及一些函数命令，使用这些命令，将大大改善编程环境。

【思考题】

1．关系运算符能否作用于矩阵或数组？矩阵和数值进行比较的实质是什么？

2．关系运算和逻辑运算有什么用途，能否配合使用？

3．想得到一个已知矩阵中所有大于特定数值的元素组成的新矩阵，应该如何操作？

4．请读者自学 find 函数用法，并求出 find 函数作用于 x = [11　0　33　0　55]'的结果。答案（ans =[1　3　5]'）

练习 12　选择结构和循环结构

知识背景

在数学计算或工程数据处理中，有时我们需要先编制特定程序，然后利用程序来完成任务。如果我们遇到较为复杂的问题，那么，在编制程序过程中可能还会遇到循环和嵌套结构。事实上，在 C 语言和 Fortran 语言里我们早已接触过类似问题，但 MATLAB 在这方面还是有一些自身的特点。熟练掌握 MATLAB 中的选择结构和循环结构是今后遇到复杂问题，建立模型，进而编程解决问题的必要条件。

主要内容

【本练习考查知识点】

本练习我们学习 MATLAB 中的选择语句 if 和 switch，要注意语句的语法结构和 end 的用法。for 和 while 用于循环结构，分别用于循环次数已知和未知的情况。使用时应注意与 end 的对应关系。

练习过程

（1）选择结构

首先我们来看一下 if 语句，MATLAB 中的 if 语句用于选择结构，它一般的形式如下：

```
  if        A1        % 表达式 1
            B1        %  命令 1
elseif      A2        % 表达式 2
            B2        %  命令 2
else
            B3        %  命令  3
end
```

从 if 语句结构中我们就能看出 if 语句执行的特点。若表达式 A1 成立，则执行命令 B1，若表达式 A2 成立，执行命令 B2；如果上述情况均未出现，则执行命令 B3。其中 elseif 语

句可以重复使用。

if 语句最简单的形式是：

```
if        A              % 表达式
          B              %  命令
end
```

有了上面的知识，我们就可以来编写一个小程序。比如说，我们要判断输入数据正负性，可以在命令区内输入：

```
a=input('a=');if   a>0 judge='正数',
elseif   a<0 judge='负数',
else   judge='零',
   end
```

回车后，首先在命令区里出现“a=”的提示符，要求输入一个数据。我们输入−5，回车后，得到的运行结果如图 12-1 所示。

上面的程序中，我们用到了 input 语句，这个语句要求用户输入数据。当我们输入一个数据后，程序就会判断这个数据的正负性，给出结果。在输入这个程序时，一定要注意，应该使照图 12-1 所示的格式输入，即“if a>0 judge='正数',”语句应跟在“a=input('a=');”之后，否则，MATLAB 将认为 a 是一个没有定义的变量。这一点要牢牢记使。

我们再来学习 switch 语句。switch 语句的结构如下：

```
switch    a          %读入一个语句
case     A1          %情况 1
          B1          %命令 1
case     A2
          B2
case  …
      …
otherwise           %其余情况
      Bn           %最后一个命令
```

我们来看一个判断是否下雨的小程序。

在命令区内输入：

```
a=input('a=') ;switch a case 1
disp('it is raining')
case 0
disp('I do not konw')
case -1
disp('it is not raining')
otherwise
disp('is it raining?')
   end
```

这个程序将可能性分为四种情况，不同情况下将得到不同的输出结果。回车后，出现提示符，要求输入一个数值。我们输入−1，得到的结果，如图 12-2 所示。



图 12-1　if 语句程序运行结果

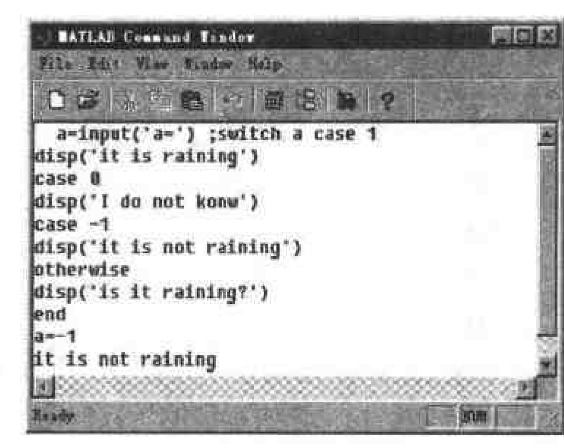


图 12-2　switch 语句程序运行结果

switch 语句根据变量或者表达式的不同取值，分别执行不同的命令。比如所举的例子中，当输入的 a 值为 1 时，即 case1，将显示“'it is raining'”，不同的 case 将执行不同的命令（输出关于是否下雨的判断）。

（2）循环结构

先来看一下 for 语句。for 语句的格式为：

```
for 变量=表达式
    命令 1
    命令 2
    …
end
```

上面所说的表达式，一般就是循环条件。for 语句一般用于循环次数已知的情况。我们来看一个具体例子。

在命令区中输入下面的程序：

```
n=5;
for i = 1:n
    for j = 1:n
        if i == j
            a(i,j) = 2;
        elseif abs([i j]) == 1
            a(i,j) = 1;
        else
            a(i,j) = 0;
        end
    end
end
  a
```

得到的结果如下：

```
a =
     2     0     0     0     0
     0     2     0     0     0
     0     0     2     0     0
     0     0     0     2     0
     0     0     0     0     2
```

这是一个 for 和 if 语句连用的例子，生成了一个主对角线元素为 2 的对角阵。请读者对照前面的 if 语句结构来分析程序的特点。请注意 end 的对应关系。每个 for 语句必须以 end 结尾，if 语句的结尾也要有 end 。我们把程序写成上面的格式是为了增加可读性，一个好的程序应该有很强的可读性，以便出错时改正及程序间转换和调用。

while 语句也是一个循环语句，它一般用于循环次数未知的情况。格式如下：

```
while        表达式
             命令
end
```

比如有这样一个问题：

请求出阶乘大于或等于 99^{99} 的最小整数。显然，我们事先无法知道要循环多少次，如果试算的话，将费时费力，这时 while 语句就派上用场了。我们编制并输入下面的程序：

```
n=1;
while prod(1:n)<99^99
      n=n+1;
end
  n
```

程序本身很简单，短短几行。我们来看一下图 12-3 所示的结果。



图 12-3 while 循环语句运行结果

从这个例子中，读者应该能够体会到 for 语句与 while 语句的不同。

大家不要小看“end”，在 if、switch、for 和 while 语句里都用到了它。end 最一般的用法是作为程序段结束的标志。使用时，要注意对应关系，不可多，不可少，不可错位。然而，end 还能作为指示符，表示向量的最后一个元素。请看下面的例子：

输入：

```
A = magic(3)
B = A(end,2:end)
```

得到的结果如下：

```
A =
    8     1     6
      3     5     7
      4     9     2
B =
    9     2
```

break 语句我们没有介绍，它一般用来跳出 for 或 while 循环。

【练习小结】

在这个练习当中，我们学习了 MATLAB 的选择结构和循环结构。if 语句和 switch 语句用于选择，根据不同情况执行不同命令，if 语句和 switch 都要以 end 结尾。 for 和 while 用于循环结构。for 一般用于循环次数已知的情况，每个 for 后都要有 end； while 一般用于循环次循未知的情况。end 是语句段的结束标志，也可作为指示符。使用时应注意 end 的对应关系。

【思考题】

1. MATLAB 中可用于编写程序的常见语句有哪些？
2. if 语句中的 end 应放在什么位置？
3. switch 语句与 if 语句有何异同？
4. 请分别使用 while 和 for 语句编写一个判断正负性的程序。
5. 使用 end 作为标识符，调出已知向量的相应元素。

练习 13　M 文件的编写

知识背景

在前面的练习中，我们主要是在命令区内输入命令（直接输入或拷贝），然后得到结果。在问题比较简单时，此方法快捷、实用。但是，当面对复杂问题和大量数据时，这样做就非常不便。在这种情况下，MATLAB 中的 M 文件显得非常有用。所谓 M 文件，就是 MATLAB 语句构成的程序文件。我们可以事先编好有用的 M 文件，然后在命令窗口内进行调用，从而轻松完成任务。

主要内容

【本练习考查知识点】

本练习我们介绍 MATLAB 中 M 文件的编写和调用。读者将详细了解到 M 文件的书写格式，使用方法及其他注意事项。另外，我们还向大家介绍拉格朗日插值程序，主要用到了 for 和 if 语句，算作对前面所学内容的复习。

练习过程

（1）命令文件

M 文件分为两类：命令文件和函数文件。它们有着不同的功能。首先我们来学习命令文件的创建。

当一个程序比较复杂，很难在短时间内调试出来时，我们一般都会把它存储起来。但存储后的调用又会产生很多问题。MATLAB 针对这种情况，增添了命令文件存储功能。有了这一功能，我们就能将函数存储为 MATLAB 能够识别和调用的文件形式，大大方便了我们的工作。

下面我们来实际创建一个 M 命令文件。

打开菜单“File”>“New”>“M-file”，新建一个编辑窗口如图 13-1 所示。这是一个还没有命名的窗口。我们在窗口输入练习 12 中建立主对角线元素为 2 的 5×5 对角阵的循环程序。

我们将这个文本 M 文件起名为 dj.m，并将它保存。保存时的界面如图 13-2 所示。



图 13-1　MATLAB 编辑窗口

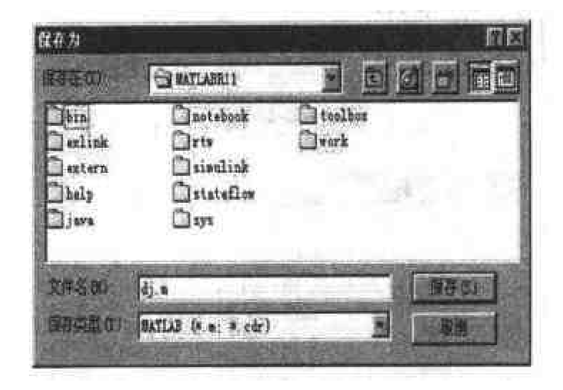


图 13-2　M 文件保存

这里我们应该注意，M 文件的命名的后缀必须是.m ，文件可以保存到任意位置，但在调用时要设好路径，否则，将会因为找不到路径而产生错误。

我们来调用这个文件，在命令区内输入：

n=4；

dj，a

但是系统给出错误信息，这是什么原因呢？这就是前面提到的路径设置的问题。我们将 dj.m 保存在 C:\MATLABR11，而 MATLAB 的默认路径不是这个名称，所以我们必须修改路径。单击“File”>“Set Path”，在弹出的路径浏览器里修改路径，如图 13-3 所示。

然后我们再来运行命令区内的命令，得到下面的结果：

```
a =
     2     0     0     0
     0     2     0     0
     0     0     2     0
     0     0     0     2
```

（2）函数文件

M 文件还有另外一种形式就是函数文件。编好的函数文件可以当作新函数一样来调用。函数文件的第一行必须是下面的格式：

function [因变量] = 函数名（自变量）

下面我们创建一个函数文件并调用。大家知道，插值方法有很多，拉格朗日方法是其中的一种，我们编写一个计算拉将朗日插值的函数 M 文件，命名为 lagr1.m，然后保存到 C:\MATLABR11。利用这个 M 文件，我们可以很轻松地计算出插值结果。M 文件窗口及程序如图 13-4 所示。

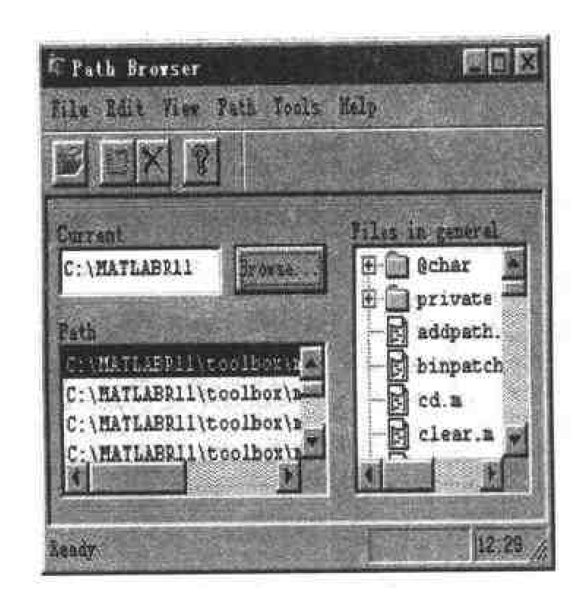
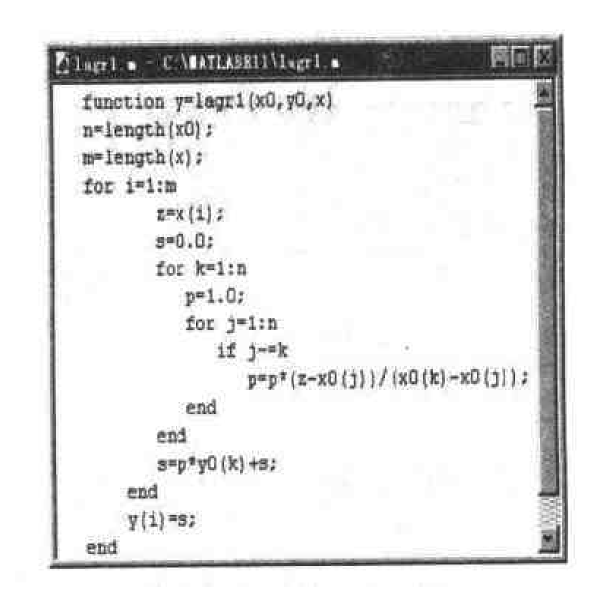



图 13-3　设置路径



```
function y=lagr1(x0,y0,x)
n=length(x0);
m=length(x);
for i=1:m
        z=x(i);
        s=0.0;
        for k=1:n
           p=1.0;
           for j=1:n
              if j~=k
                 p=p*(z-x0(j))/(x0(k)-x0(j));
           end
        end
        s=p*y0(k)+s;
     end
     y(i)=s;
end
```

图 13-4　lagr1.m 函数文件窗口

我们来分析一下 lagr1.m 文件的结构：

第一行是按照我们前面说的 MATLAB 规定的格式书写的，在这一行里，function 的字体应该是蓝色。同时，第一行规定了函数的因变量、自变量和函数名。lagr1 函数的自变量有三个，x0，y0，x。其中 x0 和 y0 是两个数组，代表对应的两组数据，x 为插值点，可以是数值，也可以是数组。如果 x 是数组，调用函数后，将输出数组。这里我们举一个例子：有两组数据如下：

x0=[0.436，0.668，0.997，1.35]；

y0=[0，0.453，2.3，4.89]；

求 x=1.25 处 y 的值。

我们在命令区里输入：

```
x0=[0.436,0.668,0.997,1.35];
y0=[0,0.453,2.3,4.89];
x=1.25
y=lagr1(x0,y0,x)
```

得到的结果如下：

```
x =
     1.2500
y =
     4.1568
```

在调用函数时，只需要输入未知量的值就可求得差值，十分方便。

在使用 MATLAB 函数 M 文件时，有几点要注意：

第一：函数 M 文件中的变量一般是局部变量，变量名独立于当前工作区和其他函数。但我们可以用 global 命令将变量设为全局变量。

第二：当 MATLAB 执行 M 文件时，首先搜索当前工作区中的变量和内建命令，然后搜索有无同名内部函数，最后按搜索路径寻找这个 M 文件。

第三：如果想要调出 M 文件，可以在命令区里输入 echo on 命令；关闭时，输入 echo off

命令。

第四：MATLAB 对函数名有和变量名一样的限制。即以字母开头；函数名由字母、数字和下划线组成。当函数文件名和函数名不同时，MATLAB 在调用时，将忽略函数名而以文件名为准。

为了说明最后一点，我们特别举一个例子，看看函数文件名和函数名不同时，将产生什么结果。

还是利用编写好的 lagr1.m 文件求插值。但我们将 M 文件中的函数名称改为“new”。当我们仍旧按照前面例子的格式采用“y=lagr1(x0,y0,x)”语句调用时，得到了和例子一样的结果。但如果我们按照 y=new(x0,y0,x)来调用时，结果如图 13-5 所示。

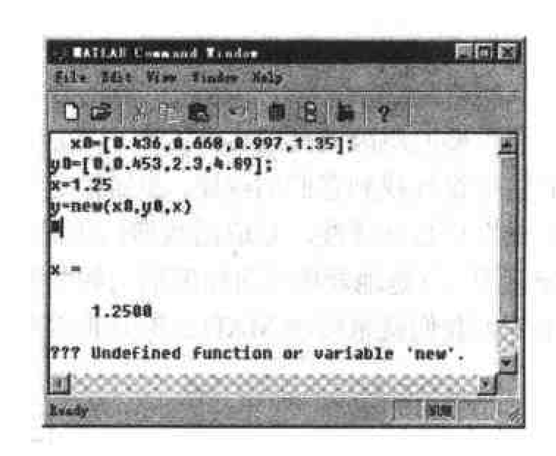


图 13-5　M 文件调入错误

系统会认为语句不合法。可见 MATLAB 中在调用 M 文件时，确实是读取文件名的。请读者自己练习求多点插值。

事实上，MATLAB 中的众多函数功能都是通过自带的函数 M 文件来实现的。每一个函数对应着一个以它自身命名的函数 M 文件。用 M 文件编辑器打开任意一个函数 M 文件，我们都可以看到相应的函数程序。在这个程序中有着相当详细的解释语句（所有%号之后的语句都是解释语句，MATLAB 不执行这部分语句），用以帮助用户了解函数的内核。从中，我们可以看出 MATLAB 是如何通过 M 文件实现相应功能的。

【练习小结】

在这个练习当中，我们学习了 MATLAB 中非常重要的部分：M 文件的编写。M 文件分为两类：命令文件和函数文件。命令文件用来存储程序或语句；函数文件用来新建函数。在编写 M 文件时要注意输入格式。调用时，要按存储的文件名来调用。并且一定要注意将文件路径设置好，否则会造成调用错误。

【思考题】

1. MATLAB 中可存储的文件形式有哪些？
2. 函数 M 文件的书写格式是什么？调用时应该注意哪些问题？
3. 在用 lagr1.m 求插值时，对输入的 x0、y0 和 x 的形式有什么要求？
4. 请编写一个判断输入数据奇偶性的程序，用 M 文件存储，并按所存储的路径调用改文件。
5. 自己查看帮助文件，调出几个 MATLAB 自带函数库里的 M 文件，看看这些函数功能是如何通过语句实现的。
6. 请比较 MATLAB 与其他数值计算语言，比如 Fortran 和 C。MATLAB 可以调用函数 M 文件进行计算，Fortran 和 C 有无相应的功能？从这一点上你觉得 MATLAB 的语言与计算特点是什么？

练习 14 MATLAB 绘 图

知识背景

MATLAB 作为一个应用广泛的数学软件，不单是因为它强大的数值计算功能，还有一个很重要的原因就是它良好的图形功能。大家都有这样的体会，面对一大堆枯燥的数据，我们往往很难找到它们的规律；而如果是图形的话，我们将非常将易的看出各个量之间的关系，是线性还是非线性，是成指数增长还是二次方关系，等等。总之，有了图形功能，将们将十分直观、清楚地表现出所研究的问题的特点，这也是科学研究和工程计算所追求的目标之一。本练习我们就来接触 MATLAB 中非常重要的部分：绘图功能。

主要内容

【本练习考查知识点】

在本练习当中，我们将学习 MATLAB 里的基本绘图命令。ploy 是一个功能较多的基本绘图命令，有好几种形式。hold 命令及用不同线型和颜色标识图形是我们将要介绍的内将。本练习用到了复矩阵，注意复矩阵的表达方法。

练习过程

（1）基本绘图命令 plot

MATLAB 中最为常用的绘制二维图形的命令是 plot ，这是一个系统自建函数，MATLAB 中的许多二维函数都建立在 plot 的基础上。利用 plot 我们就能绘出绝大多数二维图形。plot 绘图命令一共有三种形式，下面分别介绍。

plot（y）这是 plot 命令中最为简单的形式。当 y 为向量时，以 y 的元素为纵坐标，元素相应的序列号为横坐标，绘制出连线。若 y 为实矩阵，则按照列绘出每列元素和其序列号的对应关系，曲线数等于矩阵的列数。当 y 为复矩阵时，则按列以每列元素的实部为横坐标，以虚部为纵坐标，绘出曲线。曲线数等于列数。

我们在命令区中输入：

p=[0.69552,0.436,0.668,1.35]

plot（p）

将得到以 1、2、3、4 为横坐标，p 中元素为纵坐标的曲线。如图 14-1 所示。

如果输入：

x=[0+2i,0.2+4i,0.5+6i,1+7i,1.5+9i,2+12i,4+14i]

plot(x)

得到如图 14-2 所示。

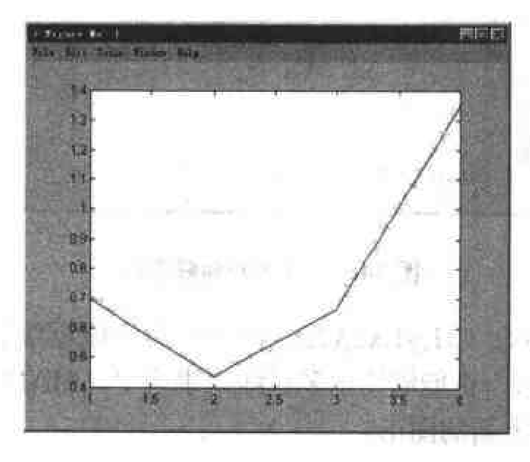


图 14-1 plot（y）命令绘图

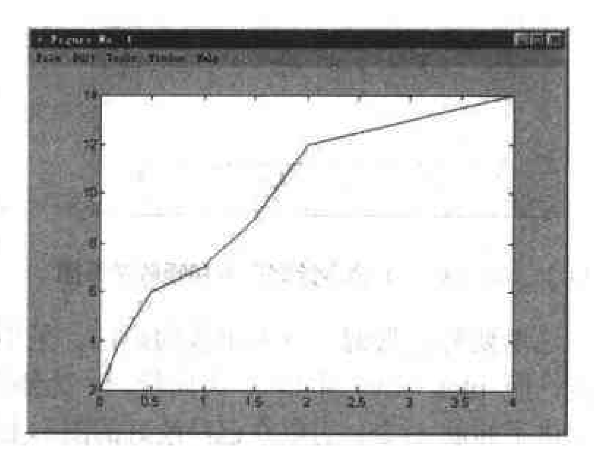


图 14-2 plot（y）命令绘制复矩阵

plot（x，y）当 x 和 y 为同维向量时，以 x 为横坐标，y 为纵坐标绘制曲线。当 x 是向量，y 是每行元素数目和 x 维数相等的矩阵时，将绘出以 x 为横坐标，以 y 中每行元素为纵坐标的多条曲线。曲线数等于矩阵行数。若 x 为矩阵，y 为相应向量，使用该命令也能绘出相应图形。

比如在命令区内输入：

A=[1.1,2,3,6];

B=[5,6,7,8;9,10,11,12];

plot(A,B)

得到图 14-3。

plot（x1,y1,x2,y2,x3,y3…） 这个命令能够绘制多条曲线，每条曲线分别以 x 和 y 为横纵坐标。各条曲线互不影响。

我们知道 MATLAB 中最为核心的部分就是矩阵操作，关于矩阵的性质，我们已经非常熟悉了。那么，我们能否利用 MATLAB 中的绘图命令来绘制一副能够直观表现出矩阵特点的图形呢？答案是肯定的。我们可以利用 plot 命令来实现这个目标。

我们绘制一个 5×5 对角阵的图形，看看有什么特点。

在命令区里输入：

z=eye(5)

plot(z)

得到图 14-4 所示的图形。从图中我们可以清楚地看到每列矩阵元素中只有一个值为 1，其余都为 0。而且我们还能从图中清楚地找到每列元素等于 1 的位置。图 14-4 确实十分的直观。

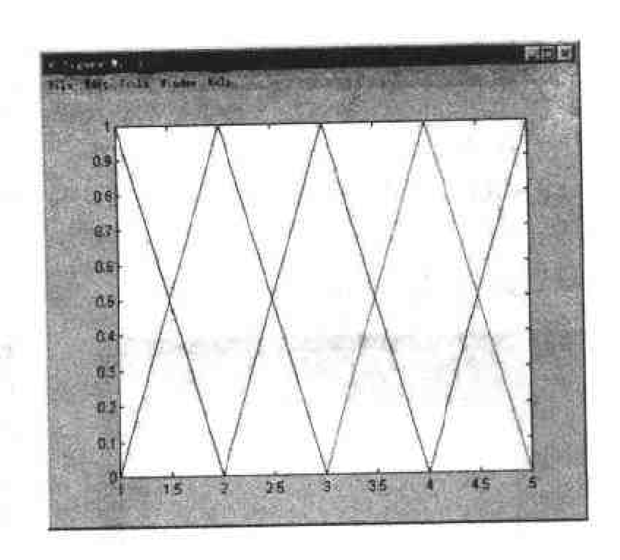


图 14-3 plot（x，y）命令绘制向量和矩阵关系图

图 14-4 5×5 对角阵图形

这里要特别强调一下多重线的画法。采用 plot（x1,y1,x2,y2,x3,y3…）命令能够绘制多重线，而 plot（x,y）中当 x 为向量，y 为矩阵时，也能够绘出多重线。事实上，MATLAB 还提供了 hold 命令。可以在已经绘好的图形上加上新的图形。

我们在命令区里输入：

```
x=linspace(0,2*pi,50);
y=sin(x);
plot(x,y)
```

得到图形如图 14-5 所示，我们可以用 hold 命令再加入一个图形。

不要关闭图形窗口，我们在命令区里继续输入：

```
hold on
z=cos(x) ;
plot(x，z)
hold off
```

将得到图 14-6 所示的结果，我们在原来的图形上加入了 z=cos(x)这个新的图形。

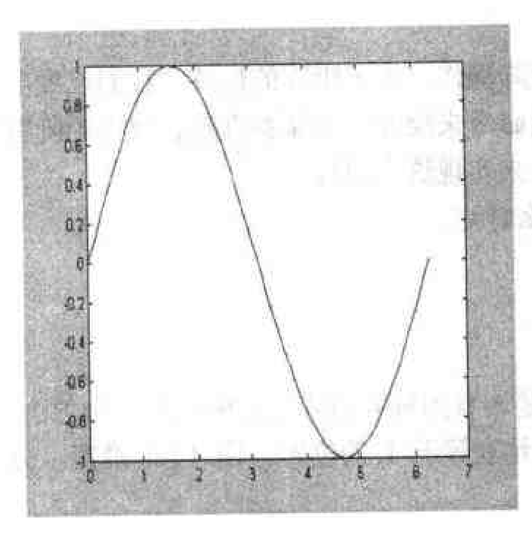


图 14-5 plot 命令绘图

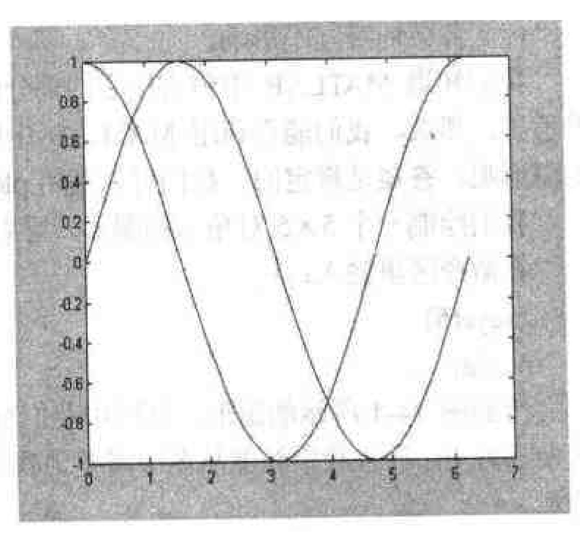


图 14-6 图形叠加

（2）线型和颜色

MATLAB 可以对线型和颜色进行设定。线型和颜色种类如下：

线：	—	实线	:	点线	—.	虚点线	——	折线				
点：	.	圆点	+	加号	*	星号	x	x 型	o	空心小圆		
颜色：	y	黄	r	红	g	绿	b	蓝	w	白	k	黑
	m	紫	c	青								

我们举一个例子，在命令区中输入：

```
x=0:pi/10:2*pi;
y=cos(x);
plot(x,y,'k+')
```

得到的图形如图 14-7 所示，请注意线型为“+”，颜色为黑色。当得到的图线较多时，采用这种标注方法，能够清晰地将各条曲线区分开，并能将特殊点明显地标识出来。

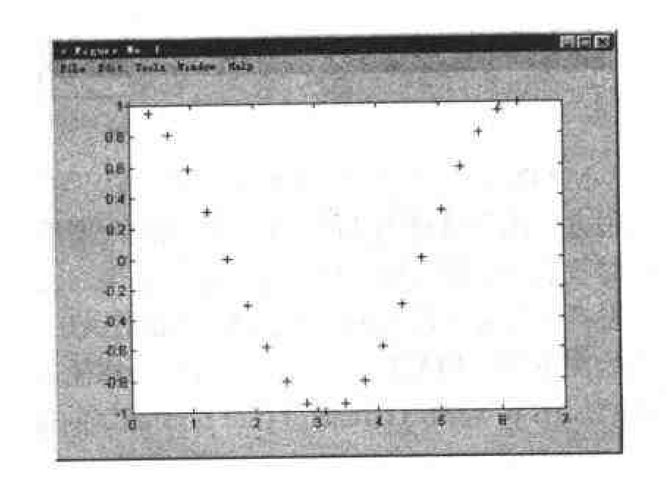


图 14-7　采用线型和颜色标注图形

从上面的例子中我们可以总结出标注的方法，即在每对数组后加一个字符串，用来说明线型和颜色。请注意，在线型和颜色之间不要加任何符号（, 或 ;），否则 MATLAB 将认为语句不合法。

【练习小结】

本练习向大家介绍了 MATLAB 中基本的绘图命令 plot 的不同形式和用法。plot 可以用来绘制多条曲线，并且可以绘制矩阵图形。hold 命令可用来在已有图形中加入新的图形，十分方便。当图线太多时，我们可以改变线型和颜色，从而达到良好的区分效果。

【思考题】

1．plot 绘图命令有几种形式？分别绘制怎样的图形？
2．请创建一个 5×5 魔方矩阵，并画出表示这个矩阵的图形。
3．绘制多条曲线的方法有哪几种？能否同时绘出两条以上的曲线？
4．画出 $y=x^2$ 的曲线（$x\in(-5,5)$）。在这条曲线上加入相同区间里 $y=\sqrt[3]{x}$ 的曲线，并且要求采用绿色折线标识。

练习 15　绘图深入学习

知识背景

MATLAB 有着良好的绘图功能，它不单可以绘制表示函数关系的图形，而且可以让我们很方便地完成对图形的操作，包括加入图形的标题，对坐标轴进行标注，绘制网格线等。同时我们还可以在图中任何地方加入文字，这在工程上有着极为重要的应用。有时候，我们还要根据实际情况绘制三维图形，MATLAB 也能够很出色地完成这个任务。正是基于强大的绘图本领，MATLAB 对其他数值计算语言有着绝对的优势。也许就是所谓“An ounce MATLAB is worth a pound of C or Fortran”的原因吧。

主要内容

【本练习考查知识点】

本练习首先介绍 MATLAB 中给图形加入网格和标记的方法。接下来介绍一些常用的二维绘图命令，最后简单介绍三维图形的绘制。本练习中，还要练习编写函数 M 文件，希望通过反复强化，达到对函数 M 文件相当熟练的程度。

练习过程

（1）加入网格和标记

我们可以在一个图形上加入网格、标题和坐标轴标记。在命令区中输入：

```
x=linspace(0,2*pi,50);y=sin(x);
plot(x,y)
grid
xlabel（'x 轴'）
ylabel ('y 轴')
title('正弦曲线')
```

得到图 15-1。

将图 15-1 与图 14-5 进行比较，发现多了网格线，坐标轴注出了“x 轴”和“y 轴”。图

形还加上了标题“正弦曲线”。这幅图看上去是不是正式多了？它是怎么实现的呢？我们回过头来分析输入的命令，得到下面的结论：

grid 放在绘图命令之后，表示给图形加上网格线

xlabel 对 x 轴进行标注，加入字符串

ylabe 对 y 轴进行标注，加入字符串

title 加入图形标题

事实上，我们还可以在图形的任意位置加上说明文字。这要用到 text 命令和 gtext 命令。得到刚才的图形后，若在命令区里输入 text（3,4,'正弦曲线'），符在坐标（3,4）处加入字符串“正弦曲线”。若输入 gtext（'正弦曲线'），则在图形上出现十字线。拖动鼠标使十字线交点到合适的地方，单击鼠标，就会将字符串“正弦曲线”放到交点处。

我们来看一个例子。在命令区里输入：

```
x=linspace(0,2*pi,50);y=sin(x);
plot(x,y)
xlabel('x 轴')
ylabel ('y 轴')
gtext('正弦曲线')
```

将出现图 15-2 所示的图形。我们可以看到十字线，拖动交点到图示位置，单击鼠标，就完成了标注，非常方便。

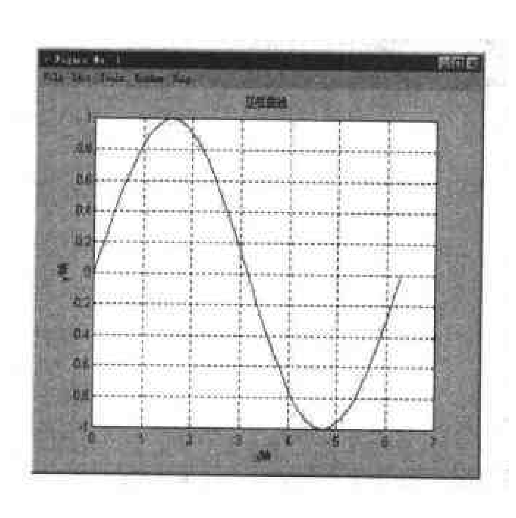


图 15-1 图形标注

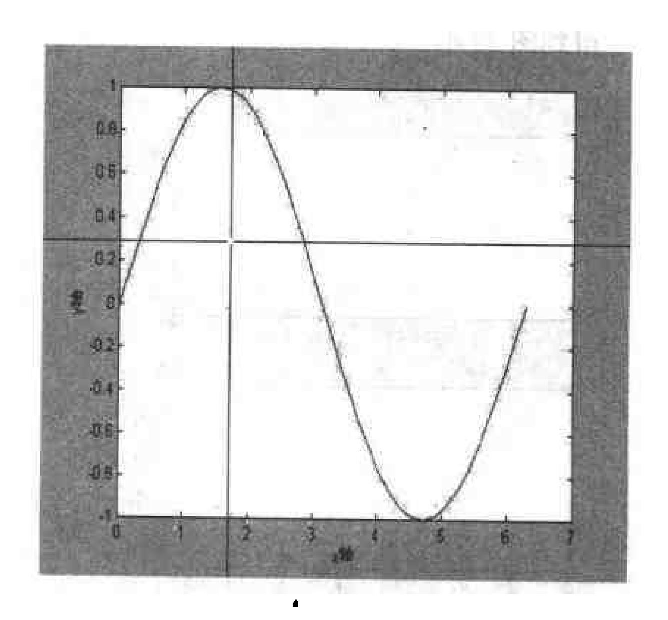


图 15-2 gtext 标注图形

（2）绘制多幅图形

我们可以在一个图形窗口里绘制多幅图形。使用这样的绘图方式，将增加对比性。要实现此功能，就要用到 subplot 命令。subplot 应写在绘图命令 plot 之前，用分号隔开。subplot 的格式是：

subplot（m,n,p）m 表示将图形窗口分为 m 行；n 表示将图形窗口分为 n 列；总共将窗口分为 m×n 部分。p 表示将绘制的图形是窗口中的第 p 部分。

我们在命令区里输入：

```
x=linspace(0,2*pi,50);
y=sin(x);
z=cos(x);
subplot(2,1,1);plot(x,y)
subplot(2,1,2);plot(x,z)
```

得到图 15-3。

（3）其他二维图形命令

fplot 是一个很有用的绘图命令。它的格式如下：

fplot（'function'，limits）：function 是将要调用的函数，也可以是函数 M 文件名。limits 即对作图所做的限制。可以是对 x、y 轴坐标范围的限制。

我们来看一个例子，首先我们创建一个函数 M 文件如下，命名为“myfun.m”，并将其保存在 C:\MATLABR11 目录下。M 文件见下。

```
function Y = myfun(x)
Y(:,1) = 200*sin(x(:))./x(:);
Y(:,2) = x(:).^2;
```

然后我们在命令区输入：

```
fplot ( 'myfun' ,[－20 20] )
```

得到图 15-4。

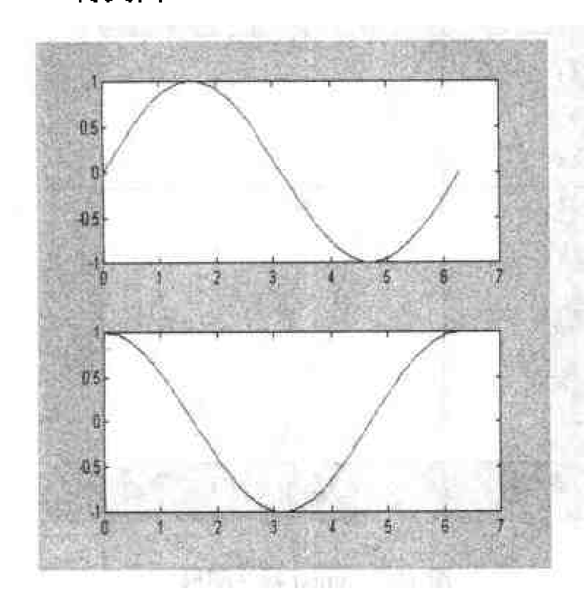


图 15-3　绘制多幅图形

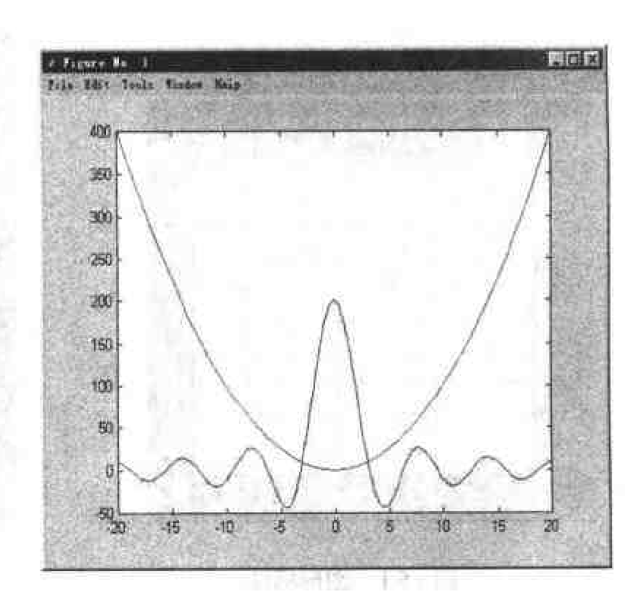


图 15-4　flot 命令绘图

另外，二维图形绘制还有如下命令：

semilogx（x,y）	半对数坐标，x 轴为常用对数坐标
semilogy（x,y）	半对数坐标，y 轴为常用对数坐标
loglog（x,y）	全对数坐标
pause	用在 plot 命令之间，暂停命令执行，直到敲击任意键。方便查图

（4）三维图形

MATLAB 提供了绘制三维图形的功能，这使得很多几何模型清晰地展现在我们眼前。下面我们来简单介绍三维图形的绘制。在命令区里输入：

```
z=peaks (25) ;
surf (z) ;
colormap (jet)
```

得到图 15-5。

这是一个带有网格的三维彩色图形。MATLAB 自带的例子

我们再看一个例子，在工作区中输入：

```
x =－2:.2:2; y = －1:.2:1;
[xx,yy] = meshgrid (x,y);
zz = xx.*exp(－xx.^2－yy.^2);
[px,py] = gradient(zz,.2,.2);
quiver(x,y,px,py,2)
```

将得到如图 15-6 所示的源汇流图形。

我们还可以用 plot3 命令绘制三维线图。格式为：

plot3（x,y,z）　当 x、y、z 为向量时，将以三个向量中相应的元素为 x 坐标、y 坐标和 z 坐标，绘出数据点，然后将这些点连接起来，得到一条空间曲线。当 x、y、z 为同维矩阵时，将分别抽出列向量绘制多条空间曲线。

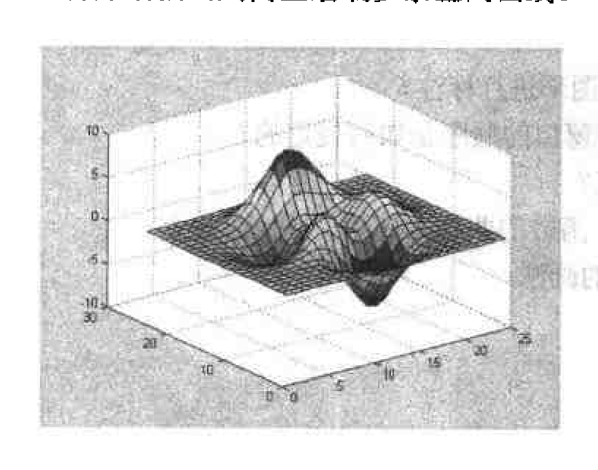


图 15-5　三维图形

图 15-6　源汇流三维图形

我们来绘制一条空间曲线。在工作区中输入：

```
p=0：pi/10：20*pi;
x=cos(p);y=sin(p);z=p;
plot3(x,y,z)
```

得到如图 15-7 所示的三维曲线。

从图 15-7 我们可以看到一条立体感非常强的三维曲线，这条曲线的形状、趋势、范围等特点非常明显地被反映出来，使人一目了然。

在进行数值计算或用 MATLAB 完成其他工作任务时，合理使用 MATLAB 的绘图功能，将取得非常好的效果。特别是三维图形，有时很难用常规方法绘出图形，但在 MATLAB 中

却可以很好地被绘制出来。读者要注意这一点。

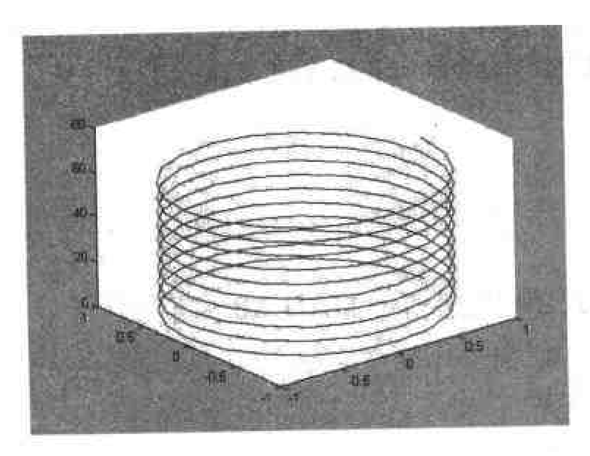


图 15-7 三维曲线

【练习小结】

本练习和上个练习相比，更加深入地介绍了 MATLAB 中的绘图命令。我们可以利用 grid 命令完成网格标记，xlabel 、ylabel 、title 和 text 命令完成图形标记。subplot 命令可以完成在同一个图形窗口中绘制多幅图形的任务。fplot 是很常用的绘图命令，希望牢记。plot3 用来绘制三维曲线。

【思考题】

1. 请思考网格线有什么作用。为什么要对图形进行标注？
2. subplot 将图形窗口分成若干部分，这些窗口的顺序是如何规定的？
3. gtext 命令与 text 命令相比，有什么优点？
4. 画出横坐标在（−15，15）上的 y=sinx 函数的曲线。应该使用什么命令？
5. 通过 MATLAB 联机帮助，学习等高线的画法。

练习 16　常微分方程（一）

知识背景

求解常微分方程在工程技术中运用比较广泛，在水力学、电工电子技术、环境模型、生态模型以及经济管理等各个领域都有重要的运用。由于常微分方程在工程技术中运用的广泛性、重要性，我们分两个练习来重点讲解和练习。

求解常微分方程，有时候会遇到无法求出解析解的情况，即不能通过等式左右同时积分求不定积分的方法求出。但是我们有时候在给定条件下，可以求出常微分方程的数值解来逼近数值解，以完成对模型的研究。

求解常微分方程数值解的方法通常有欧拉法和龙格一贝塔法，对于微分方程，欧拉方法解法的基本思想是在小区间[x(n),x(n+1)]上用差商代替方程的导数，并通过把小区间不断地划分求极限，从而最终讲到数值解。龙格－贝塔方法的基本思想也是用差商来代替导数，只不过龙格－贝塔在小区间内是运用了微分中值定理，在小区间内多取点而取加模平均值来构造精度更高的计算式。而在 MATLAB 中，主要采用龙格－贝塔法来计算常微分方模的数值解。

主要内容

【本练习考查知识点】

本练习主要通过几个解一阶常微分方程或其模型来辅导读者掌握解决常微分方程数值解的基本方法。本练习主要用到的函数是 ode23 和 ode45，其中 ode45（4 阶及 5 阶）是首选的方法，而 ode23（2 阶及 3 阶）是比 ode45 低阶的函数。其主要表达式是：

```
[t,x]=ode23('f',t0,tf,x0,TOL)
[t,x]=ode45('f',t0,tf,x0,TOL)
```

f 是将所求解的常微分方程写成的函数并存于 m 文件中，to、tf 分别为自变量的初始值和最终值，而 x0 则是函数的初始值，TOL 是误差范围（缺省值是 0.000001）。这些知识点请读者熟记。

练习过程

（1） 先以一个简单的例子来说明解常微分方程的解法：

例：求解常微分方程：$x' = -x^3$。

其中初始值为:$x'(0) = 1$

这是一个较简单的微分方程，可以通过在等式两边求积分从而求出其表达式，为了后面研究方程的数值解，这里我们采用求数值解的方法，在求解中我们只给出在区间[0,1]的数值解及图形，函数 yfun.m 文件：

```
function exer=yfun(t,x)
exer=－x.^3;
```

程序如下：

```
[t,x]=ode('yfun',[0 1],1);
plot(t,x,'k+',t,x,'r')
```

结果如图 16-1、图 16-2 和图 16-3 所示。

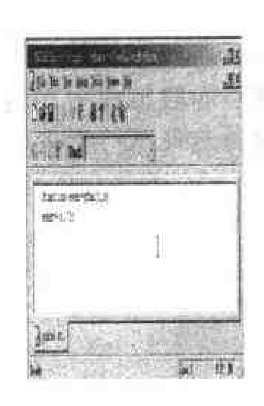


图 16-1　函数 yfun.m 文件

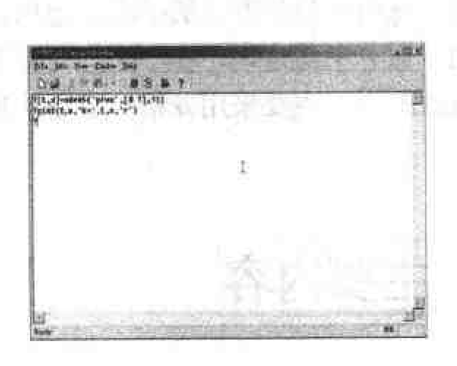


图 16-2　程序命令图

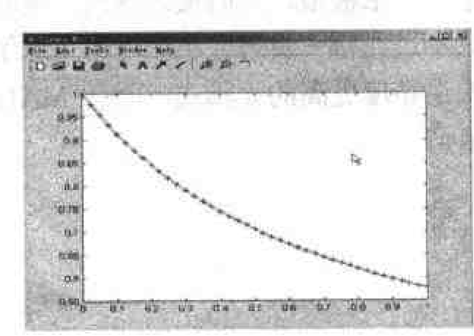


图 16-3　运行结果图

本例仅通过一个简单的练习让读者熟悉和理解求解微分方程数值解时所用的基本方法和基本函数，下面我们通过在生态学中一个重要的捕食模型来继续微分方程数值解的研究。

（2）求解二阶以上的常微分方程数值解，通常是将表达式迭代的过程，即将方程在函数中叠代表达，从而在命令行中编程实现。

我们以一个常见的物理模型为例，来说明二阶以上常微分方程的一般解法。在单摆模型中，以垂直方向为平衡位置，即 x=0 时，以右边为正方向建立摆角 x 的坐标系。在物体（通常是实心小铅球）摆动过程中，摆动的任何一个位置 x，小球所受重力沿运动轨迹方向的分力为－mgsinx，利用牛顿运动第二定律有：

$$mlx'' = -mg\sin x$$

设初始偏离角度为 a，此时无初速，则方程的初始条件为：x(0)=a,x'(0)=0。由于方程没有解析解，故我们求其数值解。初始角度为 0.1745（弧度）时，求数值解如下：

先将所列的方程存入一个函数文件中，然后再在命令区中计算调用函数文件。函数文

件如下：

```
high.m
function xexer=high(t,x)
g=9.8;
length=25;
xexer=[x(2);－g/length*sin(x(1))];
```

在命令区中输入的程序如下：

```
ts=[0,10];
x0=[0.1745,0];
[t,x]=ode('high',ts,x0)
```

运行结果为：

t	x
0	0.1745
0.0007	0.1745
0.0015	0.1745
0.0022	0.1745
0.0030	0.1745
0.0066	0.1745
0.0103	0.1745
0.0140	0.1745
0.0177	0.1745
0.0362	0.1745
0.0546	0.1744
0.0731	0.1743
0.0915	0.1742
0.1838	0.1734
0.2761	0.1719
0.3683	0.1699
0.4606	0.1673
0.7106	0.1576
0.9606	0.1440
1.2106	0.1269
1.4606	0.1068
1.7106	0.0840
1.9606	0.0591
2.2106	0.0329
2.4606	0.0058
2.7106	−0.0214
2.9606	−0.0482
3.2106	−0.0737

```
    3.4606   -0.0974
    3.7106   -0.1187
    3.9606   -0.1372
    4.2106   -0.1523
    4.4606   -0.1637
    4.7106   -0.1711
    4.9606   -0.1743
    5.2106   -0.1734
    5.4606   -0.1681
    5.7106   -0.1588
    5.9606   -0.1457
    6.2106   -0.1289
    6.4606   -0.1091
    6.7106   -0.0866
    6.9606   -0.0619
    7.2106   -0.0358
    7.4606   -0.0087
    7.7106    0.0185
    7.9606    0.0453
    8.2106    0.0710
    8.4606    0.0949
    8.7106    0.1165
    8.9606    0.1353
    9.2106    0.1508
    9.4606    0.1626
    9.5955    0.1674
    9.7303    0.1709
    9.8652    0.1733
   10.0000    0.1744
```

【练习小结】

本练习主要介绍了一阶以及二阶常微分方程的解法，并介绍了 MATLAB 中解常微分方程常用的函数及其用法实例。通过实例介绍了求解二阶常微分方程的迭代算法，并介绍了其在 MATLAB 中的实现。

【思考题】

1．请运用本练习介绍的算法以及函数来解常微分方程：

$x' = x^2$，$x'(0) = 1$

2．求解二阶常微分方程：

$$x''=-x^2+x+1,\quad x'(0)=1, x(0)=0$$

3．ode23 和 ode45 的精度有什么差别？

4．如何求解二阶以及二阶以上的常微分方程？

练习 17　常微分方程（二）

知识背景

常微分方程常用在对模型的分析运用中，特别是在环境、生态、人口等模型中可以比较方便地通过研究模模来得到一些信息并可以采取一些措施。下面我们介绍一个比较有名的生态学的人口动力学模型，可以认为是单一的捕食者和模捕食者模型。通过这个模型的研究我们可以推广到对多数模型的研究，读者可以通过这个模型来了解 MATLAB 对数学模型的处理。

主要内容

【本练习考查知识点】

本练习主要通过已有的函数来研究数学模型，并通过模型的抽象化以及对模型的研究处理让读者掌模对一般数学模型的理解，并让读者能够初步掌握常微分方程建模的基本方法。

练习过程

（1）x1 表示被捕食者，例如鸡等；x2 表示捕食者，例如黄鼠狼等。而在自然界中，各种种群是相互依存，互相制约的关系。种群 1 靠自然资源来生存，种群 2 靠模食种群 1 来生存，它们这种生存方式在生态学中称为捕食者－被捕食者模型。

（2）对于被模食者而言，其食物主要是天然资源，即被模食者主要从周围环境中取得食物，其生存与自然环模的关系比较紧密。周围环境如天气、温度、水和食物生长都会影响到捕食者的生存。而模捕食者的数量除了与自然环境有关以外，还受捕食者的影响。模食者数量少时，对被模食者的数量有一定影响。但当模食者数量激增后，其数量的增加势必要影响到被捕食者的数量。以模捕食者为食物源的捕食者数量增加使模捕食者数量减少，如果捕食者数量持续增加，则被捕食者的数量减少势必会影响到捕食者的数量，此时亦造成捕食者数量的减少，这里它们之间是一个作用与反作用的关系。

（3）对于捕食者而言，其主要食物是被捕食者（如鸡），影响其生存的主要是与其生存

相关的气候及其他因素。而其食物源是影响其数量的最大因素，捕食者数量的增加会导致被捕食者数量的减少，而当捕食者数量不足时，被捕食者数量的增加也会使捕捕食者食物减少从而反作用与被捕食者的数量。

（4）通过分析捕食者和被捕食者之间的关系，我们有如下的关系方程组：

x1'=x1-0.18x1*x2　　　　(1)

x2'=-x2+0.028*x1*x2　　　　(2)

x1(0)=30　　　　(3)

x2(0)=20　　　　(4)

这个微分方程组只是较简化地捕绘出捕食者与使捕食者之间的关系，只是一个简化捕型。我们通过赋予其一定的初值来研究这个模型。编程序如下：

biomodel.m 文件：

```
function biofun=biomodel(t,x)
biofun=[x(1)-0.01*x(1)*x(2);-x(2)+0.02*x(1)*x(2)];
```

程序如下：

```
ts=0:0.1:20;
x0=[30,20];
[t,x]=ode45('biomodel',ts,x0);
[t,x]
plot(t,x),grid,gtext('x(1)'),gtext('x(2)')
plot(x(:,1),x(:,2)),grid,xlabel('x1'),ylabel('x2')
```

运行结果如下：

t	x1	x2
0	30.0000	20.0000
0.1000	32.5109	19.2634
0.2000	35.2557	18.6517
0.3000	38.2534	18.1633
0.4000	41.5237	17.7987
0.5000	45.0870	17.5610
0.6000	48.9645	17.4559
0.7000	53.1712	17.5007
0.8000	57.7312	17.7043
0.9000	62.6726	18.0732
1.0000	68.0140	18.6292
1.1000	73.7648	19.4087
1.2000	79.9247	20.4633
1.3000	86.4842	21.8596
1.4000	93.4243	23.6788

……

9.3000	139.9776	52.0400
9.4000	146.0903	62.7493
9.5000	150.6406	76.3904
9.6000	152.9848	93.5791
9.7000	152.3984	114.9429
9.8000	148.2745	140.5770
9.9000	140.3422	169.8691
……		
17.0000	154.5240	109.6173
17.1000	151.1838	134.7472
17.2000	143.9508	163.8932
17.3000	132.9289	195.7177
17.4000	118.8590	227.8864
17.5000	103.0537	257.3525
17.6000	86.9685	281.4540
17.7000	71.7286	298.7313
17.8000	58.2099	308.3795
17.9000	47.0407	310.2391
18.0000	38.2113	305.5663
18.1000	31.2409	296.3016
18.2000	25.7959	283.8343
18.3000	21.5948	269.2821
18.4000	18.3872	253.5578
18.5000	15.9140	237.4037
18.6000	13.9719	221.3315
18.7000	12.4232	205.6908
18.8000	11.1957	190.6687
18.9000	10.2766	176.3024
19.0000	9.5869	162.7225
19.1000	9.0626	150.0046
19.2000	8.6729	138.1536
19.3000	8.3944	127.1563
19.4000	8.2113	116.9815
19.5000	8.1110	107.5901
19.6000	8.0848	98.9394
19.7000	8.1265	90.9850
19.8000	8.2313	83.6824
19.9000	8.3959	76.9875
20.0000	8.6184	70.8561

绘制的图形如图 17-1 和图 17-2 所示。

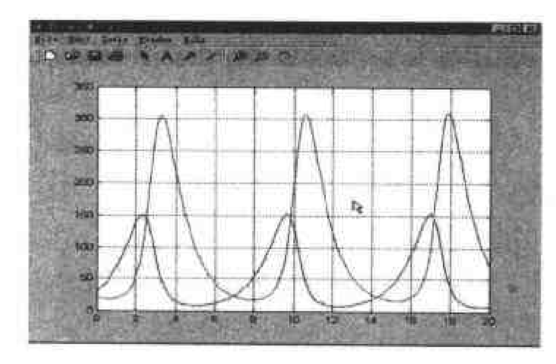




图 17-1 捕食者模型周期图

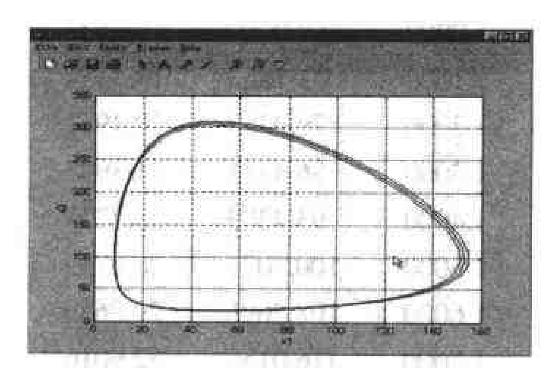


图 17-2 捕食者模型相图

通过图我们可以看出，x1 和 x2 都是周期函数，而相图(x1,x2)则趋于封闭曲线，从数值解上我们可以大致作出判断周期为 7.3，x1 和 x2 的最大、最小值都可以通过所列数据得出，并根据所列数据和周期求一个周期内的平均值。由于拟合插值前已述及，这里不再给出求解过程。而模型方程的系数都有一定的实际意义，其具体内容请参看《数学模型》（姜启源著，高教出版社）。此时，我们可以给出捕食者模型的相图。

（5）以上例题的模型被简化，如果我们加入其他因素的影响，比如说捕食动物和被捕食动物的生老病死等，我们可以得到一个以下的模型：

x1'=x1－0.1*x1*x2+0.01*t (1)

x2'=－x2+0.02*x1*x2+0.04*t (2)

x1(0)=30 (3)

x2(0)=20 (4)

依上例，编 biomodel1.m 文件如下：

```
function biofun=biomodel(t,x)
biofun=[x(1)－0.01*x(1)*x(2)+0.01*t;－x(2)+0.02*x(1)*x(2)+0.04*t];
```

在命令区中输入运行程序：

```
ts=0:0.1:20;
x0=[30,20];
[t,x]=ode45('biomodel1',ts,x0);
```

运行结果如下：

t	x1	x2
0	30.0000	20.0000
0.1000	32.5110	19.2636
0.2000	35.2559	18.6525
0.3000	38.2538	18.1651
0.4000	41.5244	17.8018
0.5000	45.0881	17.5659
0.6000	48.9660	17.4631
0.7000	53.1729	17.5106
0.8000	57.7331	17.7174

0.9000	62.6746	18.0903
1.0000	68.0160	18.6507
1.1000	73.7663	19.4354
1.2000	79.9253	20.4961
1.3000	86.4831	21.8996
1.4000	93.4203	23.7277
1.5000	100.7071	26.0769
1.6000	108.2696	29.0659
1.7000	116.0156	32.9016
……		
9.6000	149.8762	97.3544
9.7000	148.7965	118.6957
9.8000	144.2545	144.1030
9.9000	136.1076	172.7254
10.0000	124.6733	202.9964
10.1000	110.8044	232.5735
10.2000	95.7603	258.7825
10.3000	80.8213	279.4098
10.4000	67.0586	293.0548
10.5000	55.0868	299.5086
10.6000	45.0963	299.5358
10.7000	37.0540	294.1821
10.8000	30.7096	284.7402
10.9000	25.7032	272.5704
11.0000	21.7994	258.6249
11.1000	18.7320	243.7358
11.2000	16.3127	228.5049
11.3000	14.4353	213.2979
11.4000	12.9971	198.4033
11.5000	11.8766	184.0800
11.6000	11.0051	170.4539
11.7000	10.3308	157.6026
11.8000	9.8196	145.5557
11.9000	9.4470	134.3103
12.0000	9.1884	123.8592
12.1000	9.0282	114.1759
12.2000	8.9543	105.2288
12.3000	8.9575	96.9813
……		
18.1000	29.1777	268.4449

18.2000	24.8688	256.3274
18.3000	21.4400	242.9595
18.4000	18.7400	228.8936
18.5000	16.5935	214.6367
18.6000	14.8794	200.5432
18.7000	13.5313	186.8151
18.8000	12.5049	173.5666
18.9000	11.7107	160.9674
19.0000	11.1034	149.0772
19.1000	10.6497	137.9260
19.2000	10.3248	127.5192
19.3000	10.1124	117.8397
19.4000	9.9975	108.8653
19.5000	9.9704	100.5650
19.6000	10.0237	92.9050
19.7000	10.1518	85.8502
19.8000	10.3509	79.3641
19.9000	10.6190	73.4092
20.0000	10.9557	67.9480

绘制的图形如图 17-3 和图 17-4 所示。

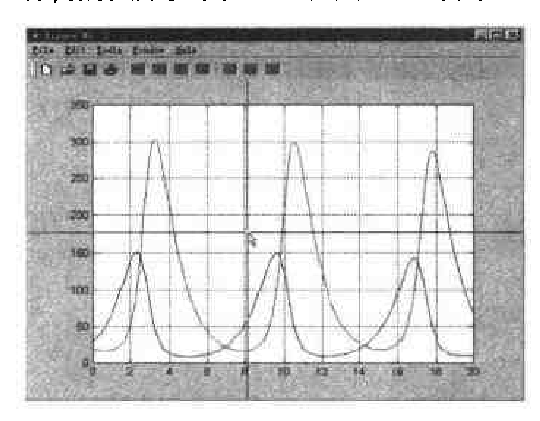


图 17-3　考虑时间因素的捕食者模型周期图

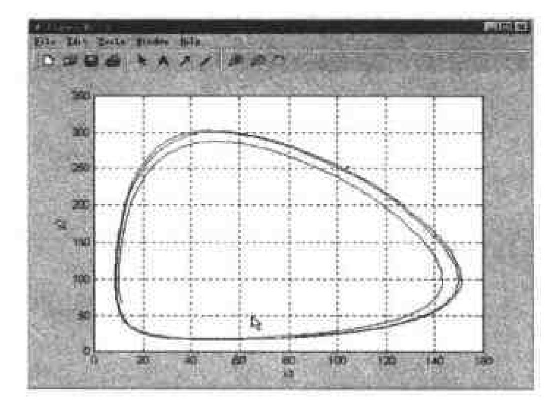


图 17-4　考虑时间因素的捕食者模型相位图

通过结果图我们可以看出相图（x1,x2）成为不断向内收敛的曲线，其具体分析留给读者。

【练习小结】

本练习通过建立在生态学上普遍使用的捕食者 —— 被捕食者模型，用常微分方程来研究模型，并在一定初始条件下来研究模型。主要讲述的是模型建立以及模型分析，并通过在 MATLAB 模拟模型来研究模型中各个因素之间的关系。

【思考题】

1. 读者参考有关书籍后，试建立有些和捕食者 —— 被捕食者有关的模型来研究捕型中各个因素之间的关系。

2. 在 MATLAB 中将本练习中介绍的各个实例分别加以实现，并改变初始条件来加以研究这些模型。

练习 18　函数的零值

知识背景

在数学运算、工程实际运用以及经济核算中，经常要求某一表达式的零值以及最大最小值，通常的方法包括解方程求解零值，配方、求导求解最大最小值。在求解零值的过程中，我们可以通过零值的求解来确定某一方程的解。

主要内容

【本练习考查知识点】

本练习旨在通过训练读者掌握 MATLAB 中求解零值和最值的基本方法，并通过函数命令和图形来直观地显示结果。本练习介绍函数“fzero”，通过练习要求读者熟悉并最终熟练使用函数命令“fzero”。命令“fzero”的使用方法是 fzero(function,x0,TOL)，functiou 为方程表达式所在的函数文件，x0 为初始值，TOL 为误差限。“fzero”采用的是叠代的方法来求解零值，使得初始估计值接近于零值。

练习过程

（1）求解函数的零值，我们先用一个简单的例子来熟悉命令“fzero”。

例：求解函数

$$f(x)=x^3-3x+3$$

的零值。

我们首先将函数表达式写入一个函数文件中，后再在命令框中调用函数“fzero”，函数文件 tzero.m 为：

```
function zexer=tzero(x)
zexer=x.^3－3*x+3;
```

在命令区中输入程序，求解零值：

```
result=fzero('tzero',−0.3);
reulst=fzero('tzero',0.3);
span=−5:0.1:5;
plot(span,tzero(span)),grid
```

得到的图形如图 18-1 所示。

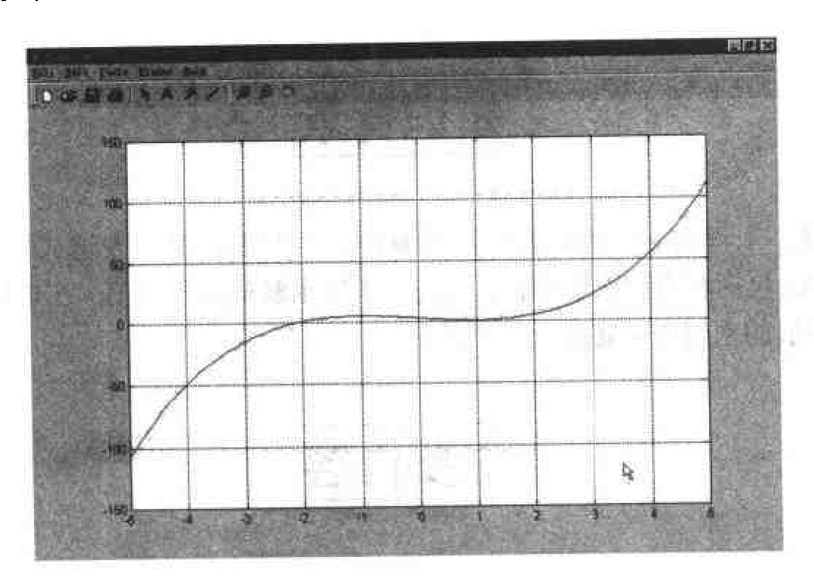


图 18-1 被求解函数图

程序运行结果为：

Zero found in the interval: [1.8722,−2.4722].

ans=−2.1038

我们可以用这种方法求出已知方程的根，这种方法简便快捷。缺点是由于要试初始值，故有可能漏掉根。

（2）我们看稍复杂一些的函数

例：求解函数 的零值。

同理有函数文件：

```
function zexe=tzero1(x)
tzero1=sin(x)−2*x+2;
```

在命令区运行命令：

```
reulst=fzero('tzero1',0.4)
```

运行结果为：

Zero found in the interval: [−0.112, 0.912].

reulst=0.8354

通过作图命令，得到图 18-2。

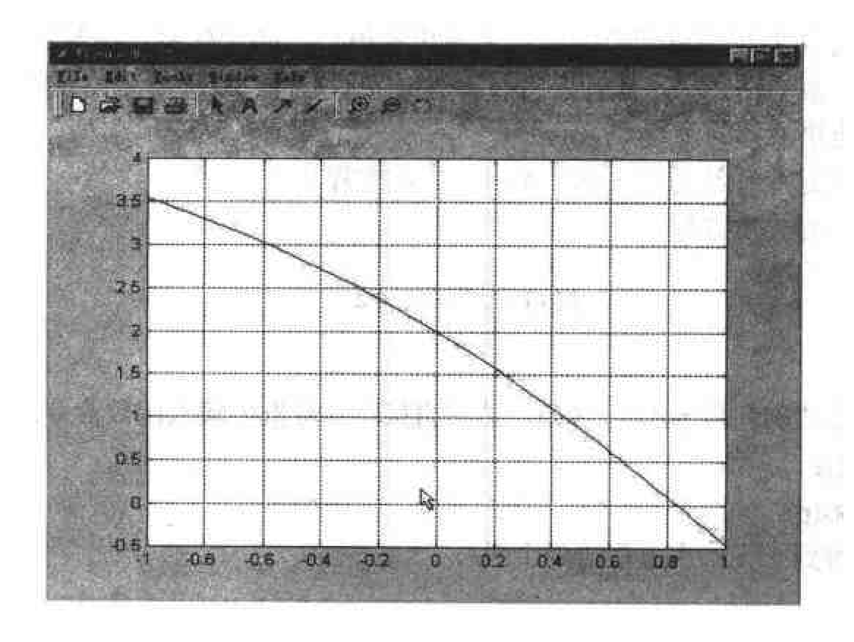


图 18-2 被求解函数图

通过这种方法我们可以比较方便地求出方程的解，无论方程多复杂，可以将方程的函数表达式写入文件，而后调用命令来求解。

（3）我们再来看一个比较复杂的例子。在这个例子中，将有二次项、分式的混合表示式，在某些点上将会造成分母为零的情况，此时函数在此次点没有取值。

例：求解函数：

$$f(x)=\frac{3x-4}{(x-1.2)^2+0.02}+\frac{4x-7}{x^3+2}+\frac{5x}{(x-0.4)^2+6}+\frac{3x}{x^3+1}$$

的零值点。

我们首先来分析这个问题。对于有的函数在某些区间内的零值点可能不同，所以通过尝试不同的初始值可以得到不同的零值点。

我们在命令区里输入函数及命令程序：

tzero3.m 文件：

```
function exercise=tzero3(x)
  exercise=(3*x－4)/((x－1.2).^2+0.02)+(4*x－7)/(x.^3+2)+5*x/((x－0.4).^2+6)+3*x/(x.^3+1);
```

在命令区中输入：

```
result=fzero('tzero3',1)
result=fzero('tzero3',－1.2)
result=fzero('tzero3',－2)
```

程序运行结果为：

```
result=1.3140
result=－1.2599
result=－2.5342
```

由于这种方法要运用试初值的方法来求出零值点，必须先要对函数的零值点所在区间给出一个估计。故当估计不足或者给出的初值范围不当时，有可能漏掉零值点。我们下面介绍一个求零值点更方便的函数“roots”。

（4）我们在这个小练习中先解一个一元二次方程。

例：求解一元二次方程

$$f(x)=x^2-3x+2$$

的根。

先建立表达式的向量 p=[1 －3 2]，然后直接在命令框中输入函数命令：

```
p=[1 -3 2];
results=roots(p)
```

运行可以得到一元二次方程的两个根：

```
results =2
         1
```

（5）我们还可以求更一值的多项式的根。

例：求解多项式

$$f(x)=x^4-3x^3+x-2$$

的根。

先建立多项式表达式的向量 p1=[1 －3 0 1 －2]，接着在命令框中输入函数命令求解：

```
p1=[1 -3 0 1 -2];
results=roots(p1)
```

运行结果为：

```
results =
         2.9630
        -0.9061
         0.4716+0.7229i
         0.4716-0.7229i
```

这个四次方程有四个根，在本例中为两个实根两个虚数根。

【练习小结】

本练习主要介绍了求解函数零值常用的两个函数“fzero”和“roots”。函数“zero”需要通过尝试不同的区间来确定函数零值的范围，函数“roots”可以求出多项式的零值。求函数表达式的零值点时，如果表达式比较复杂，则可以用函数命令“fzero”。但是需要输入初始值在一定区间进行迭代计算，而且在某些时候会漏掉某些区间的零值点。可以通过作图来直观地研究函数的零值点防止漏解。在一般的多项式求根或求零值点时，可以使用命令“roots”。

【思考题】

1．用本练习中介绍的函数分别求解函数：

$$f(x)=x^3+(x-0.98)^2/(x+1.25)^3-5(x+\frac{1}{x})$$

的零值点。

练习 19 最 值 问 题

知识背景

求解最大值最小值是数学计算以及一些经济核算中的重要组成部分，最大最小值的一般求法是求导进行计算判断。先介绍极值点的求法，如果函数在 N(m,n)上可导，在点 m 处连续，如果在点 m 两侧 f'(x)变号，则点 m 为 f(x)的极值点；当 x<m 时，f'(x)>0(<0),x>m 时，f'(x)<0(>0)，则 m 为极大(小)值点。求出在极值点处的函数值即为函数的极值。

当函数 f(x)在闭区间 I=[a,b]上连续时，必在 I 上取得最大和最小值，最大值和最小值可能是端点 a 或 b，也可能是区间中的任意内点。如果最值点是区间内点时，必为函数 f(x)的极值点。因此，只需要找出(a,b)内的全部驻点(f'(x)=0 的点)和非驻点(f'(x)不存在的点)，比较这些点的函数值以及端点函数值的大小，即可以求出区间[a,b]的最大最小值。

主要内容

【本练习考查知识点】

在 MATLAB 中求最大最小值的方法是进行迭代计算，即先选定一个初始值 x_0，假设满足最值要求的自变量为 a。从 x_0 开始跌代得到 x_1，再继续跌代得到 x_2、……、x_t，迭代的终值应满足：$|f(a)-f(x_t)|$充分小，即小于一定的误差范围。从初始值跌代到一定精度后停止，一般应使$|x_0-a|$的值达到一定较度后即满足要求。本练习主要练习使用两个函数命令'fmin'和'fmins'，前者是求单变量函数的最小值，后者是求多变量函数的最小值，开始时应该给一个初始值，这个初始值为数组。

练习过程

（1）我们先求一个比较简单的例子。

例 1：求函数

$$f(x)=x^4-2x^3$$

在区间[-1,2]的最小值。

先预编一个函数文件 yminis.m，把函数表达式存于这个文件中。编好的 yminis.m 文件

见下：

```
function exer=yminis(x)
exer=x.^4－2*x.^3;
```

在命令区中键入：

```
minnum=fmin('yminis',－1,2);
ts=－1:0.1:2;plot(ts,yminis(ts))
```

可得运行结果为：

```
minnum =
        1.5000
```

我们可以得到图 19-1 进行直观的分析：

从图 19-1 上我们可以看到最低点处的位置，故运行结果是正确的。

（2）对于更复杂一些的函数，我们也可以采取同样的办法。

例 2：求函数

$$f(x)=\frac{4x+3}{(x-1.4)^2+3}+\frac{2x+1}{x^3+2}+\frac{4x+5}{(x-0.9)^2+1}$$

在区间[-2,3]的最小值。

同样，我们有 yminis1.m 文件：

```
function exer=yminis1(x)
exer=(4*x+3)/((x－1.4).^2+3)+(2*x+1)/(x.^3+2)+(4*x+5)/((x－0.9).^2+1);
```

在命令区中输入命令和参数有：

```
minnum=furin('yminis1',－2,3);
```

运行结果为：

```
minnum =－1.2599
```

此时我们应该注意最小值和极值的区别，函数给出的是区间中的极小值，需要求最值还应该比较驻点和非驻点以及端点的函数值大小。

（3）MATLAB 命令中没有求最大值的函数命令，但是可以利用加负号求最小值的方法求最大值。我们可以求上两例子的最大值：

例：求（1）中的最大值。

函数文件 ymax.m

```
function exer=ymax(x)
exer=－x.^4+2*x.^3;
```

在命令区中键入命令：

```
maxnum=fmin('ymax',－1,2)
```

得到结果为：

```
maxnum=－1.0000
```

所以此时最大值应为：－maxnum=1.0000。输入绘图命令，得到函数图值如图 19-2 所示。

```
ts=－1:0.1:2;
plot(ts,ymax(ts))
```

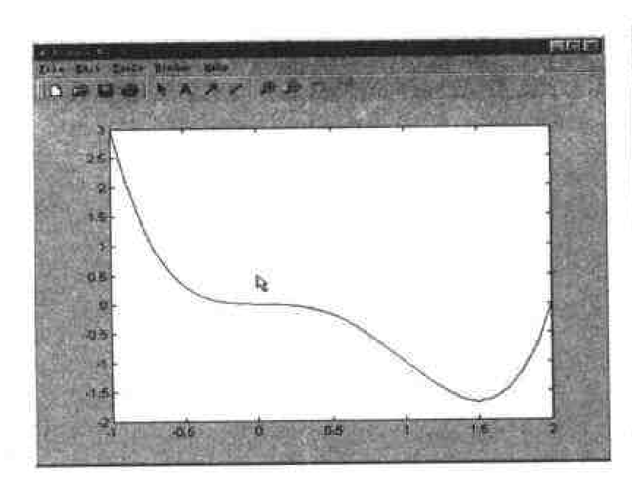


图 19-1　被求函数图像

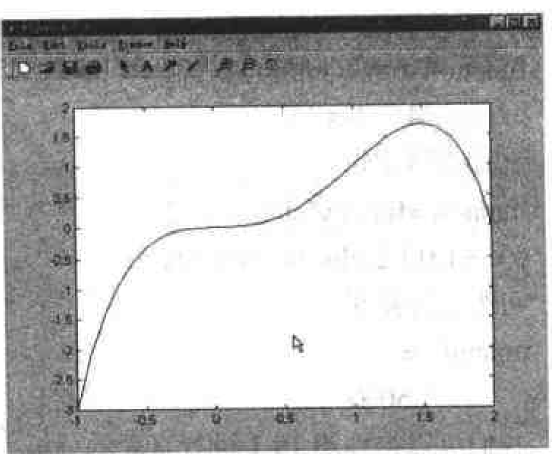


图 19-2　函数图像

从图像可以看出，此时最大值从端点处取得。

例：求(2)中的最大值。

函数文件 ymax1.m：

```
function exer=ymax1(x)
exer=-(4*x+3)/((x-1.4).^2+3)-(2*x+1)/(x.^3+2)-(4*x+5)/((x-0.9).^2+1);
```

键入参数和命令：

```
maxnum=fmin('ymax1',-2,3)
```

运行结果为：

```
maxnum =
        1.1914
```

故此时最大值应该为−maxnum=−1.1914。

（4）我们还可以求出多元函数的极值，这时我们用的是函数命令“fmins”，用法与函数命令“fmin”一样。

例：求函数

$$f(x_1,x_2)=x_1^2+x_2^2-2x_1+x_2-2$$

的最小值。

建立函数文件 ymin.m，存入所求的函数表达式：

ymin.m 文件：

```
function temp=ymin(x)
temp=x(1).^2+x(2).^2-2*x(1)+x(2)-2;
```

键入函数命令，并赋予初始值：

```
minnum=fmins('ymin',[1,0])
```

结果为：

```
minnum =
1.0001    -0.5000
```

即在（1.0001,−0.5000）处取得最小值。作出图像研究，可以从图 19-3 上看出最小值。

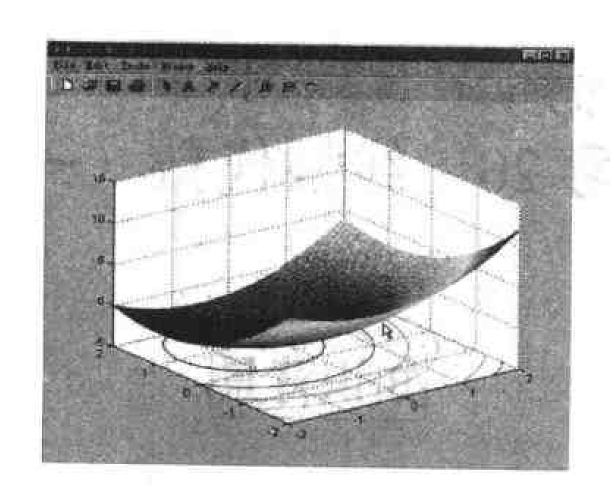


图 19-3　被求函数的图像

【练习小结】

本练习主要介绍了函数最值和函数极值求法的基本思想，通过迭代来求函数极值和最值是 Matlab 的基本思想。首先选定初值，通过迭代不断地逼近所要求的极值和最值，满足一定精度则停止迭代。在 Matlab 中用到的函数主要是“fmin”和“fmins”，“fmin”主要求单目标函数(一元函数)的极值和最值，而“fmins”则可以求出多元函数的极最值问题。

【思考题】

1. 求一元函数

$f(x)=x^3+2x^2+x+-5$

的最小值和最大值，并作出图像比较。

2. 求一元函数

$f(x)=\dfrac{3x+1}{x^2+1}+\dfrac{2x-6}{x^3-2}+\dfrac{x+1}{(x-1.63)^2+1.2}$

的最大值和最小值

3. 求函数

$f(x_1,x_2)=x_1^2+2x_2^2-3x_1+x_2-3$

的最小值。

练习 20　线性代数（一）

知识背景

MATLAB 具有很强的矩阵预算功能，我们将花比较多的篇幅来介绍关于矩阵部分的内容。我们将由浅入深地介绍关于矩阵的一些基本概念以及相关的函数命令。

主要内容

【本练习考查知识点】

本练习首先介绍一些基本的关于矩阵运算的基本函数命令，并通过实例来介绍这些函数命令。通过本练习，让读者掌握矩阵的基本运算以及能够使用相关函数命令来进行计算。设 A 为一个方阵：

（1）则函数 det(A)为求方阵 A 的行列式的值，如果 A 不为方阵，将给出错误信息。

（2）函数 rank(A)为求方阵 A 的秩(矩阵 A 中线性无关的行数和列数)。

（3）函数 inv(A)为求方阵 A 的逆矩阵。但是当 A 是奇异矩阵或者近似为奇异矩阵时，系统会给出错误信息。

（4）函数 pinv(A)为求矩阵 A 的伪逆。当 A 是 m×n 的矩阵时，则伪逆大小为 n×m。

（5）函数 trace 为求矩阵 A 的迹，即对角线元素之和。

（6）函数 orth(A)为求 A 的正交基，列数等于矩阵 A 的秩。

（7）函数 null(A)为求 A 的零空间的正交基，它的列数等于零空间的维数。

（8）函数 subspace(A,B)是求由矩阵 A 和 B 的列划分的子空间的夹角，列的长度必须相等。

练习过程

已知矩阵为：

A=[3 5;4 7]，B=[1 3 6;2 5 8;4 6 9]，C=[1 4 9; 2 3 5]；

① 求行列式的值，在命令区中键入函数命令：

```
det(a),det(b),det(c)
```

运行结果为：

ans=−9

ans=1

由于 c 不是方阵，故系统给出信息：

??? Error using ==> det

Matrix must be square.

② 求值，键入

rank(a),rank(b),rank(c)

得到结果：

ans=2

ans=3

ans=2

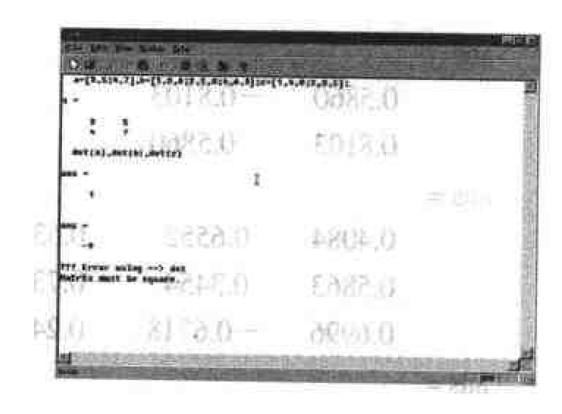


图 20-1　行列式求值运行图

③ 求逆矩阵，输入

inv(a),inv(b),inv(c)

得到结果：

ans=

7	−5
−4	3

ans=

0.3333	−1.0000	0.6667
−1.5556	1.6667	−0.4444
0.8889	−0.6667	0.1111

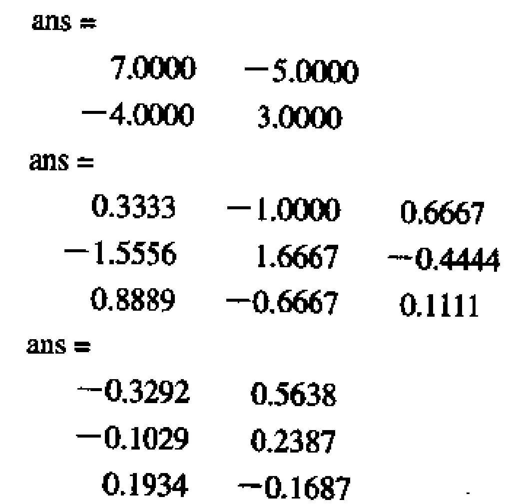
同样由于 c 不为方阵，所以系统给出错误信息：

??? Error using ==> inv

Matrix must be square.

④ 求矩阵的伪逆，有

pinv(a),pinv(b),pinv(c)



ans =

7.0000	−5.0000
−4.0000	3.0000

ans =

0.3333	−1.0000	0.6667
−1.5556	1.6667	−0.4444
0.8889	−0.6667	0.1111

ans =

−0.3292	0.5638
−0.1029	0.2387
0.1934	−0.1687

⑤ 求矩阵的迹，有

```
trace(a),trace(b),trace(c)
```

运行结果为：

```
ans=10
ans=15
ans=4
```

⑥ 求矩阵的正交基，则

```
orth(a),orth(b),orth(c)
```

运行结果为：

```
ans =
       0.5860    -0.8103
       0.8103     0.5860
ans =
       0.4084     0.6552     0.6355
       0.5863     0.3454    -0.7328
       0.6996    -0.6718     0.2431
ans =
       0.8524     0.5229
       0.5229    -0.8524
```

⑦ 求矩阵的零空间，则

```
null(a),null(b),null(c)
```

运行得：

```
ans =
   Empty matrix: 2-by-0
ans =
   Empty matrix: 3-by-0
ans =
     0.4491
    -0.8340
     0.3208
```

Empty matrix 表示零空间是空的，即得到的是零向量。

⑤ 求矩阵的夹角，则

```
subspace(null(a),orth(a'))
```

结果为：

```
ans =1.5708
```

所得结果为 1.5708，说明这两个空间是正交的。

【练习小结】

本练习首先介绍了最基本的矩阵概念和 MATLAB 中用于矩阵运算的命令，通过几个实例让读者熟悉这些函数，并通过练习能够掌握这些基本的命令。本练习涉及到的函数命令较

多，可视作对本书的一部分矩阵问题的补充和复习。

【思考题】

1．复习本练习中介绍过的矩阵函数。

2．已知矩阵 M=[1 2 5;2 3 7;7 5 9]，N=[1 4;5 3]，P=[1 2 4;2 5 4]，Q=[1 2;3 6;7 4]，分别用本练习中介绍过的矩阵函数来求这些矩阵的相关值。

3．函数命令 orth 用于什么地方？使用时应注意哪些问题？

练习 21　线性代数（二）

数学知识背景

在机械、土木、电子、化工、生态学、核物理、气体力学、弹性力学、微分方程等学科中，求特征值和特征向量是一个重要的内容。设 A 是 n 阶方阵，若存在数 a 以及非零向量 X 使得 AX=aX，则称 a 是矩阵 A 的特征值，称 X 是属于特征值 a 的特征向量。

主要内容

【本练习考查知识点】

本练习介绍一般求特征值的函值，以及在一些特定条件下求解特征值和特征向量所用的分解方法。设 A 是矩阵，则命令函数 eig(A)为求矩阵 A 特征值所形成的向量。我们还将介绍计算特征值和特征向量过程中所用到的 QR 分解和 QZ 分解。设矩阵 A 为 m×n 矩阵，QR 分解所用的命令为“qr”函数命令 qr(A)特产生一个 m×m 的方阵和一个 m×n 的上三角阵，可以先用 hess(A)命令产生一个 hess 矩阵。在 QR 分解计算中，有一个迭代的过程，即算法为：用 hess 命令求出矩阵 A 的 hess 矩阵，让 Ak=A(初始时 k=0)，找到 Ak 的分解，Ak=Qk*Rk。用迭代计算下一个矩阵，A(k+1)=Qk*Rk，通过迭代计算最终可以特到主对角线上是特征值的矩阵。

练习过程

（1）下面我们先用一个简单的实特来熟悉函数命令的用法：

例：求矩阵 A= [1 0;－2 －1]的特征值和特征向量。

直接在命令区中我入命令：

```
a=[1,0;－2,－1];
eig(a)
```

得到结果为：

```
ans =
    -1
     1
```

由此可知矩阵 A 有两个特征值，结果可如图 21-1 所示。

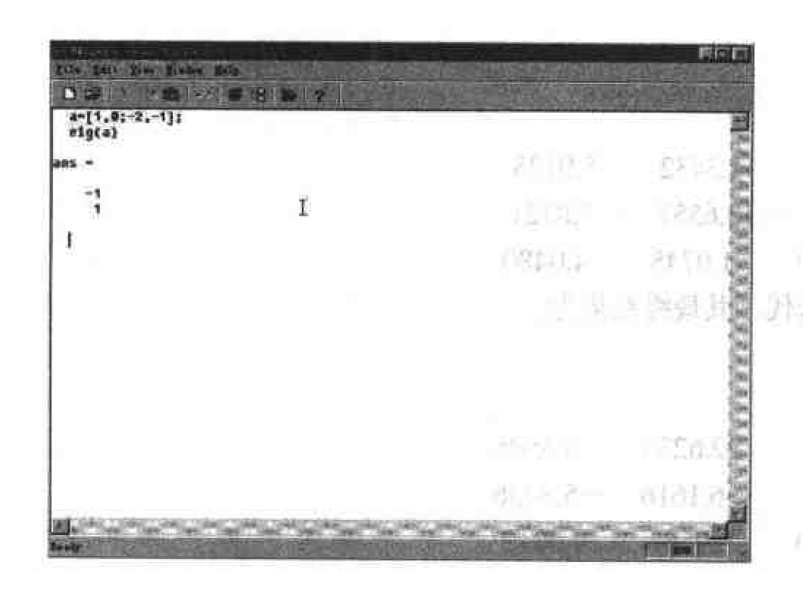


图 21-1　求矩阵的特征值

再由：

```
[m,n]=eig(a)
```

可得：

```
m =
          0    0.7071
     1.0000   -0.7071
n =
    -1     0
     0     1
```

其中 m 为对应相应特征值的特征向量构成的矩阵，而 n 为主对角线上由特征值组成、其余各位上是零的矩阵。

（2）我们用 QR 分解矩阵的方法来计算特征值。

例：求矩阵 A=[−9 −3 −16; 13 7 16; 3 3 10]的特征值和特征向量。

在命令区中输入命令并进行迭代可得：

```
a=[-9,-3,-16;13,7,16;3,3,10];
a1=hess(a);
[m1,n1]=qr(a1);
a2=n1*m1
```

运行可得 a2 为：

```
a2 =
```

```
    1.7992   26.8770  -12.6126
    2.3625    4.5085   -0.1434
         0    4.9518    1.6923
```

接着进行迭代：

```
[m2,n2]=qr(a2);
a3=n2*m2
```

得到结果：

```
a3 =
   17.6077   11.3432    5.0128
  -15.3516  -13.6557   -5.8721
         0    1.0748    4.0480
```

接着进行迭代，其最终结果为：

```
……
a11 =
   10.1297   22.6238   15.3505
   -0.0924   -6.1616   -5.8036
         0    0.0562    4.0319
```

由于矩阵 A 实际的特征值为 10，−4 和 6，对角线上的数在迭代到一定数量之后可以与实际特征值接近。

迭代算法程序运行如图 21-2 所示，程序运行的最终结果如图 21-3 所示。

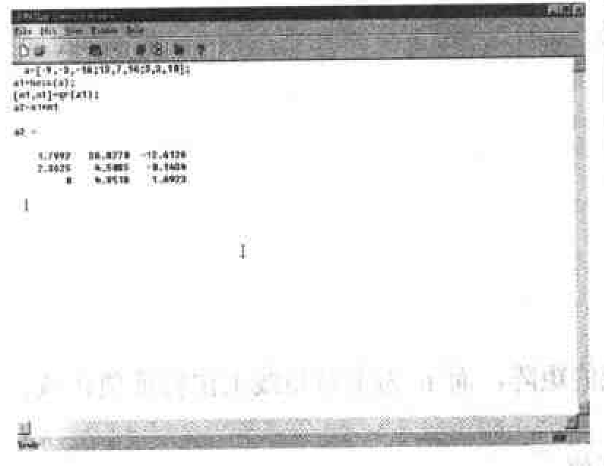


图 21-2　迭代算法程序

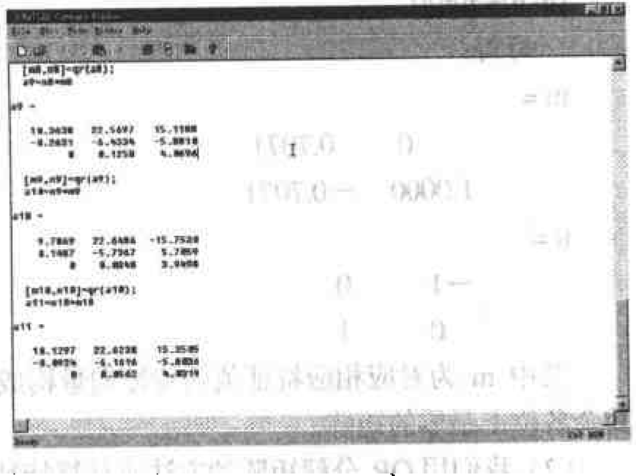


图 21-3　迭代结果图

【练习小结】

本练习介绍了用函数命令来求解特征值和特征向量，以及如何用 QR 分解算法来求特征值的方法。通过实例练习，希望读者能够掌握求特征值和特征向量的方法。

【思考题】

1. 矩阵 A=[1 4 3;2 5 4;3 6 8]，求 A 的特征值。
2. 已知矩阵 P=[1 4 7 3;5 6 3 8;3 6 7 3;1 8 4 2]，求 P 的特征值和特征向量。
3. 已知矩阵 M=[4 3 7;3 2 6;8 7 9]，用 QR 分解法求 M 的特征值。
4. 已知矩阵 N=[1 2 4;2 4 5;6 3 5;1 7 3]，用 QR 分解法求 N 的特征值。

练习 22　线性代数（三）

数学知识背景

在线性代数中经常要求解线性方程组，求解线性方程组有很多方法，包括高斯消元法、行列式求值计算法和迭代计算法等。在求解线性方程组的过程中，通常对系数矩阵进行研究，并通过有条件分解来简化计算。迭代计算主要先给定一定初值，后进行迭代计算，再用控制条件来进行计算控制。

主要内容

【本练习考查知识点】

在本练习中，主要为下一练习求解线性方程组作准备，将介绍 LU 分解法、Cholesky 分解法、QR 分解法。

练习过程

（1）MATLAB 能随着系数矩阵的不同采用不同的方法来求解线性方程组，采用高斯消元法和部分消元法的主要思路是对矩阵进行 LU 分解，即把系数矩阵 A 分解为 A=LU，其中 U 是一个上三角阵，L 是一个带有单位对角线的下三角矩阵。

有时候我们也可以求交换矩阵 P，使 PA=LU，其中交换矩阵 P 的大小为 n×n，为一个单位矩阵。MATLAB 中用以进行 LU 分解的函数命令为“lu”：

[L,U]=lu(A)

L 和 U 分别为所求的上三角阵和交换下三角阵，且 L 是一个带有单位对角线的下三角矩阵和交换矩阵。

[L,U,P]=lu(A)

L、U、P 分别为上三角矩阵、有单位对角线的下三角矩阵和交换矩阵，满足 LU=PA。

例：求矩阵 A=[1　−2　1; 2　1　−3;　−1　1　−1]的 LU 分解。

直接在命令区中输入可得：

```
a=[1,−2,1;2,1,−3;−1,1,−1];
```

[L,U]=lu(A)

得到运行结果：

```
L =
     0.5000     1.0000
     1.0000          0
U =
     4.0000     1.0000
          0     2.5000
```

再在命令区中输入：

[L,U,P]=lu(A)

得到运行结果如图 22-1 所示。

```
L =
     1.0000          0
     0.5000     1.0000
U =
     4.0000     1.0000
          0     2.5000
P =
      0     1
      1     0
```

（2）正定对称矩阵可分解成对角元素为正的下三角阵与它的转置矩阵之积，即：A=LL(T 为上标)或者表示为 A=LDL(T 为上标)；其中 L 是单位上三角阵，D 是元素为正的对角阵。这种分解称为三角分解或者称为 Cholesky。在 MATLAB 运算中，进行 Cholesky 分解的函数是'chol'，用法为：

R=chol(A)

其中，R 是矩阵 A 的 Cholesky 因子，是一个上三角矩阵。但是如果 A 不是一个正定阵的话则给出一个错误信息。这是可以使用命令：

[R,err]=chol(A)

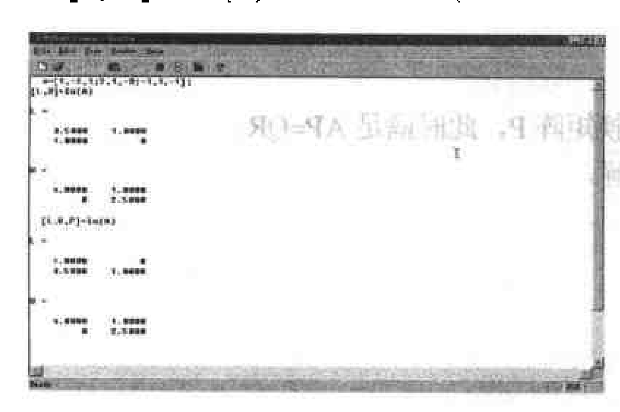




图 22-1　矩阵的 LU 分解

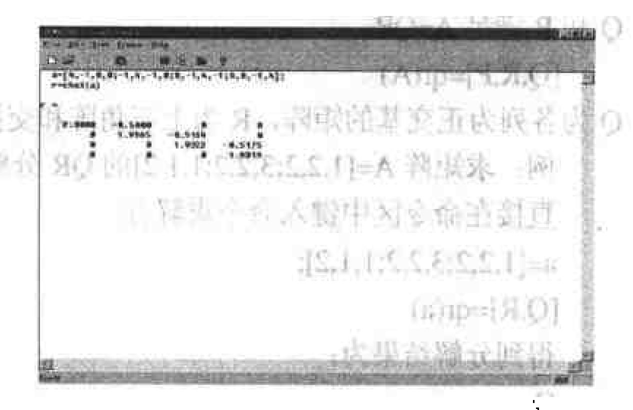


图 22-2　矩阵的 Cholesky 分解

此时，R 仍为矩阵 A 的 Cholesky 因子，当矩阵 A 不是一个正定矩阵时，则不给出错误

信息，但是 err 的值为非零。

例：求矩阵 A=[4 −1 0 0; −1 4 −1 0; 0 −1 4 −1; 0 0 −1 4]的 Cholesky 分解。

输入命令进行分解得到结果如图 22-2 所示。

```
a=[4,-1,0,0;-1,4,-1,0;0,-1,4,-1;0,0,-1,4];
r=chol(a)
```

运行得到结果为：

```
r =
    2.0000   -0.5000         0         0
         0    1.9365   -0.5164         0
         0         0    1.9322   -0.5175
         0         0         0    1.9319
```

我们可以输入命令来检验：

```
resulttest=r'*r
```

其结果为：

```
resulttest =
    4.0000   -1.0000         0         0
   -1.0000    4.0000   -1.0000         0
         0   -1.0000    4.0000   -1.0000
         0         0   -1.0000    4.0000
```

即又得到了矩阵 A，符合正定阵的定义。

（3）假设矩阵 A 是 n×n 的方阵，那么 A 可以分解为：

A=QR

其中 Q 是一个正交矩阵，R 是一个大小和 A 相同的上三角阵，因此 Ax=b 可以表示为 QRx=b 或者表示为：Rx=Qb。此时这个方程的系数矩阵是上三角的，故能比较方便地求解。进行 QR 分解的函数命令为"qr"，假设 A 为 m×n 矩阵，则：

```
[Q,R]=qr(A)
```

求得的 Q 和 R 分别为 m×m 的矩阵和上三角矩阵，而且 Q 的各列形成了一个正交基，其中 Q 和 R 满足 A=QR。

```
[Q,R,P]=qr(A)
```

Q 为各列为正交基的矩阵，R 为上三角阵和交换矩阵 P，此时矩足 AP=QR。

例：求矩阵 A=[1,2,2;3,2,2;1,1,2]的 QR 分解。

直接在命令区中键入命令求解有：

```
a=[1,2,2;3,2,2;1,1,2];
[Q,R]=qr(a)
```

得到分解结果为：

```
Q =
   -0.3015    0.9239   -0.2357
   -0.9045   -0.3553   -0.2357
   -0.3015    0.1421    0.9428
```

R =

−3.3166	−2.7136	−3.0151
0	1.2792	1.4213
0	0	0.9428

再在命令区中键入：

[Q,R,P]=qr(a)

可得结果，如图 22-3 所示。

Q =

−0.5774	0.4082	−0.7071
−0.5774	−0.8165	−0.0000
−0.5774	0.4082	0.7071

R =

−3.4641	−2.8868	−2.8868
0	−1.6330	−0.4082
0	0	−0.7071

P =

0	1	0
0	0	1
1	0	0

QR 分解可以用来求解超静定方程组，超静定方程组为方程个数多于未知变量个数的方程组。同样，QR 分解可以用来求特征值以及特征向量。

（4）将矩阵 A 分解为 A=D−L−U，其中 D=diag(a11,a12,a13,...,ann)，而矩阵 L 和 U 分别为下三角阵和上三角阵。当对角阵 D 非奇异（即有对角阵对角线上的各个元素之不等于零），则 Ax=b 可以化为：

x=D(−1)(L+U)+D(−1)b　　(1)

若记 B1=D(−1)(L+U)，其中 f1=D(−1)b，则可以将(1)的迭代形式写作：

$$X_{k+1}=B1X_k+f1 \quad (k=0，1，2，\cdots)$$

如果序列{X_k}收敛于 x，则 x 必是方程组(1)的解，即是方程组 Ax=b 的解。这种迭代方式称为雅可必迭代。如果在 D 非奇异的条件下，(D−L)可逆，则可以得到：

B2=(D−L)(−1)U，其中 f2=(D−L)(−1)b

$$X_{k+1}=B2X_k+f2 \quad (k=0,1,2,\ldots)$$

这种迭代方法称为高斯—赛德尔迭代，与雅可必迭代相比，其优点在于用新分量代替旧分量后，精度得以提高。而且在迭代过程中，可以不必用两个工作单元存放 x(k)，x(k+1)，可以只用一套工作单元。对于迭代过程中迭代值是否收敛于正确值，可以有如下判断条件：

若 A 是严格对角占优，则雅可必和高斯—赛德尔迭代均收敛；若 A 对称正定，则高斯—赛德尔迭代收敛。

在运用迭代时，主要的函数命令有：

diag(x)

输入向量 x，则输出的是以 x 为对角元素的对角阵；若输入的是矩阵 x，则输出的是 x 的对角元素构成的向量；

triu(x)

输入矩阵 x，输出为 x 的上三角阵；

tril(x)

输入矩阵 x，输出为 x 的下三角阵；

triu(x，1)

输入矩阵 x，输出为 x 的上三角阵，但是对角元素为零；

triu(x，－1)

输入矩阵 x，输出为 x 的下三角阵,但是对角元素为零；

例：分别用以上函数命令来求矩阵 A=[1,2,3;4,5,6;7,8,9]的对角阵和上、下三角阵。

在命令区中键入：

```
a=[1,2,3;4,5,6;7,8,9];
b=diag(a)
c=triu(a)
d=tril(a)
```

可得结果，如图 22-4 所示。

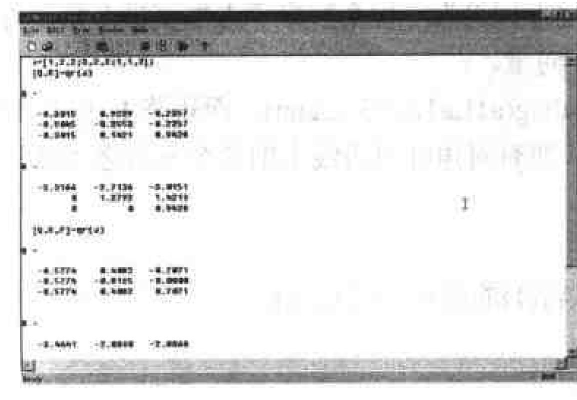


图 22-3　矩阵的 QR 分解

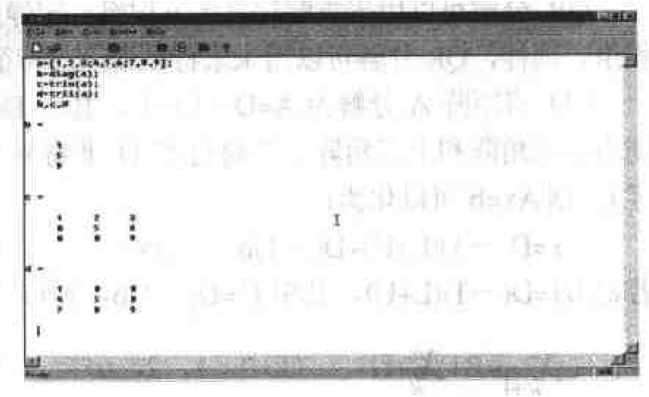


图 22-4　求矩阵上下三角阵

```
b =
    1
    5
    9
c =
    1    2    3
    0    5    6
    0    0    9
d =
    1    0    0
```

```
4     5     0
7     8     9
```

【练习小结】

在本练习中我们主要介绍了在解练性方程组中用到的一些分解方法以及迭代方法，并通过实例给出了分解方法和迭代方法的函数命令。本练习的目的是为了解线性方程准备。

【思考题】

1. 已知矩阵 M=[1,2,6;4,2,7;8,9,3]，求矩阵的 LU 分解。
2. 求上练矩阵的 Cholesky 分解。
3. 求上述矩阵的 QR 分解。
4. 求上述矩阵的对角阵、上下三角阵。

练习 23　线性代数（四）

数学知识背景

线性方程组有很多解法，上一个练习介绍了一些准备工作，本练习将详细讨论线性方程组的解法。

主要内容

【本练习考查知识点】

本练习将给出一些线性方程组解法实例，从简单的方程组到复杂的超静定方程，我们将给予详尽的讨论。

练习过程

（1）在 MATLAB 中符号“\”有很重要的功能，它既可以进行数值或变量的除法运算，同样可以进行矩阵的运算，我们用符号“\”来进行最简便的矩阵除法运算来求解方程组。

例：求解方程组

3x+2y=5　　(1)

5x－7y=29　　(2)

我们在命令区中直接进行计算即可：

```
a=[3,2;5,－7];
b=[5;29];
x=a\b
```

运行结果如图 23-1 所示。

```
x =     3
       －2
```

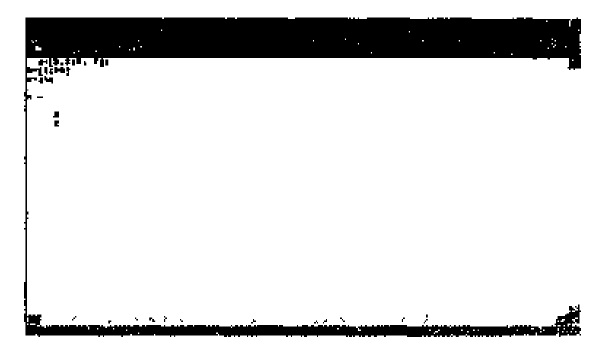


图 23-1　直接除法求解方程组

（2）用 LU 分静法来求解方程组。

例：求解方程组 Ax=b，其中 A=[1,2,• 2;1,1,1;2,2,1]，b=[1.8;0.6;3.1]。

在命令区中输入：

```
A=[1,2,－2;1,1,1;2,2,1];
b=[1.8,0.6,3.1]';
[m,n]=lu(A);
x=inv(n)*inv(m)*b;
```

运行可以得到结果为：

```
x=
        7.0000
        －4.5000
        －1.9000
```

（3）用雅可比迭代法求解方程组时，先预编函数来求解。

预编的函数为 yakebi.m

```
function x=yakebi(A,b,k,n)
L=－tril(A,－1);U=-triu(A,1);
D=diag(diag(A));B1=D\(L+U);
f1=D\b;
x=zeros(n,1);
  for i=1:k
      x=B1*x+f1;
  end
```

例：采用雅可比迭代法求解(1)中的方程组。

直接在命令区中预编的函数来计算方程组：

```
A=[1,2,－2;1,1,1;2,2,1];
b=[1.8,0.6,3.1]';
k=4;
n=3;
x3=yakebi(A,b,k,n);
```

运行可得结果为：

```
x=
        7.0000
        －4.5000
        －1.9000
```

所得结果与采用 LU 分解法求解方程组的解所得到的解一致。

（4）用高斯—塞德尔迭代法求解方程组时也要预编函数来求解方程组。

函数文件 gs.m

```
function y=gs(A,b,k,n)
L=－tril(A,－1);U=－triu(A,1);
D=diag(diag(A));
B2=(D－L)\U;
f2=(D－L)\b;
y=zeros(n,1);
```

```
    for i=1:k
        y=B2*y+f2;
    end
```

例：求解（1）中的方程组。

在命令区中直接引用函数来求解方程组：

```
A=[1,2,-2;1,1,1;2,2,1];
b=[1.8,0.6,3.1]';
k=4;
n=3;
x= gs(A,b,k,n)
```

运行后得到结果为：

```
x=
        7.0000
       -4.5000
       -1.9000
```

【练习小结】

本练习主要通过采用直接除法、LU 分解法、雅克比迭代以及高斯—塞德尔迭代法来求解方程组。一般情况下，雅可比迭代收敛得很好，而高斯—塞德尔迭代公式则有可能不收敛。具体细节请参考有关教学参考书。

【思考题】

1．已知方程组 Mx=n，其中 M=[2,−1,1;2,2,2;−1,−1,2]，n=[2.1;4.2;1.4]，请分别用直接法、LU 分解法、雅克比迭代以及高斯—塞德尔迭代法来求解方程组。

练习 24　优化计算（一）

数学知识背景

优化计算在工程技术、科学研究和经济管理等很多方面有广泛的运用，结构设计在满足强度要求等条件下使所用材料总重最轻，资源分配要使用户利用有限的资源产生的总效益最大，安排运输方案要在满足物资需求和装载条件下使运输总费用最低等等。建立优化模型，确定问题的决策变量，构造目标函数 f(x)和 x 的取值范围，在一定的约束条件下来求解。

主要内容

【本练习考查知识点】

本练习主要介绍目标函数 f(x)和 x 在一定取值范围下所能得到的最有解，本练习先不考虑约束条件。本练习主要熟悉 MATLAB 优化工具箱的基本用法，对不同算法进行初步分析、比较。

练习过程

（1）根据给出的数据，试据此拟合生产函数中的参数，如何看待用线性最小二乘法和非线性最小二乘法拟合的结果。应用柯布—道格拉斯(Cobb-Douglas)生产函数模型：$Q=a*K^{\alpha}L^{\beta}$，$0<\alpha$、$\beta<1$。以1989年为1.00。

年　份	总产值（万亿元）	资金（万亿元）	劳动力（亿人）
1984	0.7171	0.2469	4.8179
1985	0.8964	0.3386	4.9873
1986	1.0202	0.3846	5.1282
1987	1.1962	0.4322	5.2783
1988	1.4928	0.5495	5.4334
1989	1.6909	0.6095	5.5329
1990	1.8531	0.6444	5.6740
1991	2.1618	0.7517	5.8360
1992	2.6635	0.9636	5.9432
1993	3.4515	1.4998	6.0220
1994	4.5006	1.8944	6.1470

用非线性最小二乘法拟合。

编写函数youhua.m如下：

```
function y=youhua(c)
Q=[0.7171,0.8964,1.0202,1.1962,1.4928,1.6909,1.8531,2.1618,2.6635,3.4515,4.5006];
q=Q/1.6909;
K=[0.2469,0.3386,0.3846,0.4322,0.5495,0.6095,0.6444,0.7517,0.9636,1.4998,1.8944];
k=K/0.6095;
L=[4.8179,4.9873,5.1282,5.2783,5.4334,5.5329,5.6740,5.8360,5.9432,6.0220,6.1470];
l=L/5.5329;
y=q－c(1)*k.^c(2).*l.^c(3);
```

在 MATLAB 命令区中输入以下命令：

```
c0=[1,1,1];
c=leastsq('youhua',c0),
c=0.9858      0.6300      2.4290
y=sum(youhua(c).*youhua(c))      %计算误差平方和
```

计算所得的误差平方和为 y=0.0138

所以，可以得到结果：

a=0.9858　　α =0.6300　　β =2.4290

（2）对于（1）中的实例采用线性最小二乘法拟合。

利用 MATLAB 对超定方程组给出最小二乘准则下的解的功能，将式 $Q=a*K^{\alpha}L^{\beta}$ 两边取对数，得到超定线性方程组 lnQ=lua+ α *lnK+ β *lnL。

在 MATLAB 命令区中输入以下命令：

```
Q=[0.7171,0.8964,1.0202,1.1962,1.4928,1.6909,1.8531,2.1618,2.6635,3.4515,4.5006];q=log(Q/1.6909);
K=[0.2469,0.3386,0.3846,0.4322,0.5495,0.6095,0.6444,0.7517,0.9636,1.4998,1.8944];
k=log(K/0.6095);
L=[4.8179,4.9873,5.1282,5.2783,5.4334,5.5329,5.6740,5.8360,5.9432,6.0220,6.1470];
l=log(L/5.5329);
aa=[1,1,1,1,1,1,1,1,1,1,1];
A=[aa',k',l'];
b=q';
c=(A\b)';
a=exp(c(1))
```

运行可得结果为：

```
c=－0.0094      0.6208      2.3728
a=0.9906
```

所以，最终结果为：

a=0.9906　　α =0.6208　　β =2.3728

下面计算误差平方和。接着输入以下命令：

```
w=a*(K/0.6059).^c(2).*(L/5.5329).^c(3);
```

```
q=Q/1.6909;
sum=0;
for i=1:11
    sum=sum+(w(i)-q(i)).^2;
end
y=sum,
```

运行可得 y=0.0149。

非线性最小二乘拟合与线性最小二乘拟合两种方法相比，结果相差不是很大。就误差而言，用非线性最小二乘拟合法略小一点。

【练习小结】

本练习主要介绍了优化计算的一些基本方法，主要介绍用乘性最小二乘和非线形最小二乘法来求解优化问题。

【思考题】

请将本练习中的例题在 MATLAB 中实现。

练习 25　优化计算（二）

数学知识背景

实际问题一般都是有约束优化的，故在现实生活中遇到约束优化问题时，必须研究最优解位于可行域边界上时的性质。最优解要用到线形规划和非线形规划的理论，具体请参考有关运筹学方面的论著。

主要内容

【本练习考查知识点】

本练习主要讲述 MATLAB 优化工具箱的基本用法解线性规划和非线性规划的方法，和建立实际问题的线性规划和非线性规划模型。

练习过程

（1）已知函数 $f(x,y)=e^x(4x^2+2y^2+4xy+2y+1)$，初值（－1，1），在 $xy-x-y+1.5\leqslant 0$，$xy+10\geqslant 0$；约束条件下解非线性规划。

先箱写存放函数的文件，模写函数 ysyouhua1.m 如图 25-1 所示。

```
function [f,g]= ysyouhua1 (x)
f=exp(x(1))*(4*x(1).^2+2*x(2).^2+4*x(1)*x(2)+2*x(2)+1);
g(1)=x(1)*x(2)－x(1)－x(2)+1.5;

g(2)=－x(1)*x(2)－10;
```

在 MATLAB 命令区中输入以下命令：

```
x0=[－1,1];[x,opt]=constr(' ysyouhua1',x0);
x,f=opt(8),n=opt(10)
```

运行结果，如图 25-2 所示。

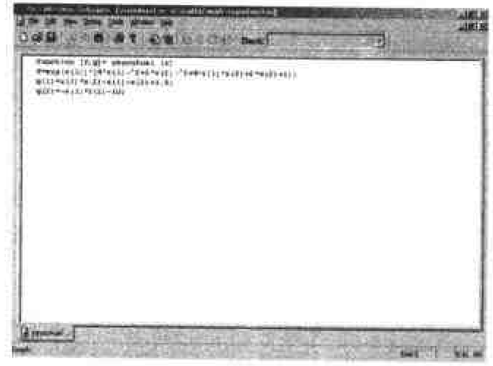
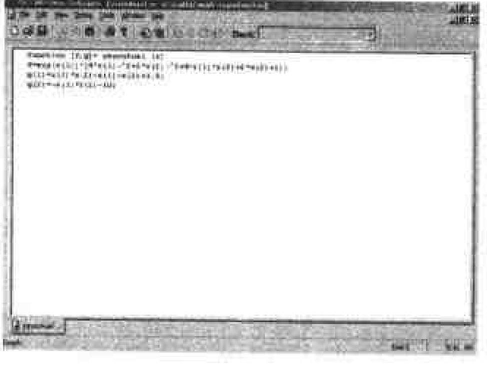



图 25-1　被求函数

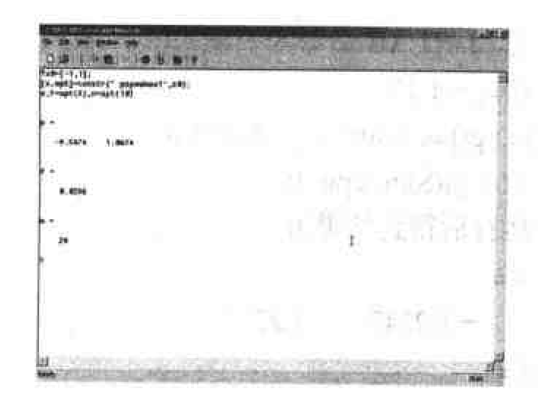


图 25-2　运行结果图

```
x=
  −9.5474
   1.0474（最优解）
f=
   0．0236（最优值）
n=
   29（迭代次数）
```

（2）对于（1）的例题，改变约束条件 xy−x−y+1.5≤0，xy+10≥0，x，y≥0 求解非线形优化。

在 MATLAB 命令区输入以下命令：

```
x0=[−1,1];
v1=[0,0];
[x,opt]=constr(' ysyouhua1',x0,opt,v1);
x,f=opt(8),n=opt(10)
```

运行后得到结果为：

```
x =
        0      1.5000
f =
     8.5000
n =
     10
```

所以 x=0，1.5000（最优解），f=8.5000　（最优值），n=10（迭代次数）。

（3）在xy−x−y+1.5≤0，xy+10≥0，x+y=0约束条件下求解非线形规划时，将函数ysyouhua1.m修改如下：

```
function [f,g]= ysyouhua2(x)
f=exp(x(1))*(4*x(1).^2+2*x(2).^2+4*x(1)*x(2)+2*x(2)+1);
g(1)=x(1)*x(2)−x(1)−x(2)+1.5;
g(2)=−x(1)*x(2)−10;
g(3)=x(1)+x(2);
g(4)=−x(1)−x(2);
```

在 MATLAB 命令区中输入以下命令：

```
x0=[-1,1];
[x,opt]=constr(' ysyouhua2',x0);
x,f=opt(8),n=opt(10)
```

运行后得到结果为：

```
x =
   -1.2247      1.2247
f =
    1.8951
n =13
```

所以 x=−1.2247，1.2247（最优解），f=1.8951（最优值），n=13（迭代次数）。

（4）在 $xy-x-y+1.5\leqslant 0$，$xy+10\geqslant 0$，用分析梯度计算求解非线形规划。

将函数ysyouhua1.m修改如下：

```
function  [f,g]= ysyouhua3 (x)
f=exp(x(1))*(4*x(1).^2+2*x(2).^2+4*x(1)*x(2)+2*x(2)+1);
g(1)=x(1)*x(2)-x(1)-x(2)+1.5;
g(2)=-x(1)*x(2)-10;
```

原函数对x的偏导数为：$e^x(4x^2+2y^2+4xy+8x+6y+1)$；

对y的偏导数为：$e^x(4x+4y+2)$。

于是编写梯度函数grad1.m如下：

```
function  [df,dg]=grad1(x)
df=[exp(x(1))*(4*x(1).^2+2*x(2).^2+4*x(1)*x(2)+8*x(1)+6*x(2)+1),exp(x(1))*(4*x(1)+4*
x(2)+2)];
dg=[x(2)-1,-x(2);x(1)-1,-x(1)];
```

在 MATLAB 工作区输入以下命令：

```
x0=[-1,1];
[x,opt]=constr(' ysyouhua1',x0,opt,[],[],'grad1');
x,f=opt(8),n=opt(10)
```

运行后结果为：

```
x =
   -9.5474      1.0474
f =
    0.0236
n =
    11
```

所以 x=−9.5474，1.0474（最优解），f=0.0236（最优值），n=11（迭代次数）。与（1）中用数值方法计算梯度相比，计算次数减少了。

【练习小结】

本练习主要介绍了求解非线形规划所用的方法，在约束条件下求解方程所用到的函数，以及采用分析梯度计算可以减少迭代次数的算法。

【思考题】

1．已知函数 $f(x_1,x_2,x_3)=\min(x_1+2x_2+x_3)$，$x_1\geq 0, x_2\geq 0, x_3\geq 0$，约束条件为：$x_1-2x_2+x_3=4, -x_1+3x_2=5$，求解其非线形规划。

2．$\min(2x_1+x_2+2x_3-3x_4), x_i\geq 0$ (i=1,2,3,4)，约束条件为：$8x_1-4x_2-x_3+3x_4\leq 10, 2x_1+3x_2+x_3-x_4\leq 7, -2x_2-x_3+4x_4\leq 12$，求约束条件下的非线性规划。

练习 26　优化计算（三）

数学知识背景

以上练习介绍了优化计算的基本理论和基本解法，在无约束和有约束条件下求解问题的基本思想，而在实际问题中求解优化问题常常是有约束条件的。

主要内容

【本练习考查知识点】

本练习将介绍优化计算在实际运用中的两个实例，通过本练习可以了解解决实际问题中优优计算的基本方法和基本思路。

练习过程

（1）例题：

炼油厂将 A，B，C 三种原油加工成甲、乙、丙三种汽油。一桶原油加工成一桶汽油的费用为 4 元，每天至多能加工汽油 14000 桶。原油的买入价、买入量、辛烷值、硫含量，及汽油的卖出价、需求量、辛烷值、硫含量由下表给出。问如何安排生产计划，在满足需求的条件下使利润最大？

一般来说，做广告可以增加销售，估计一天向一种汽油投入一元广告费，可使这种汽油日销量增加 10 桶。问如何安排生产和广告计划使利润最大？

原 油 类 别	买入价（元/桶）	买入量（桶/天）	辛烷值（%）	硫含量（%）
A	45	≤5000	12	0．5
B	35	≤5000	6	2．0
C	25	≤5000	8	3．0

汽 油 类 别	卖出价（元/桶）	需求量（桶/天）	辛烷值（%）	硫含量（%）
甲	70	3000	≥10	≤1.0
乙	60	2000	≥8	≤2.0
丙	50	1000	≥6	≤1.0

以下均按当天生产满足当天需求计算。

设生产甲、乙、丙三种汽油，各需A、B、C三种原油x1、y1、z1，x2、y2、z2，x3、y3、z3。则有如下约束条件：

① 需求量：

x1+y1+z1=3000

x2+y2+z2=2000

x3+y3+z3=1000

② 买入量：

x1+x2+x3≤5000

y1+y2+y3≤5000

z1+z2+z3≤5000

③ 辛烷值：

12%*x1+6%*y1+8%*z1≥10%*3000

12%*x2+6%*y2+8%*z2≥8%*2000

12%*x3+6%*y3+8%*z3≥6%*1000

④ 硫含量：

0.5%*x1+2.0%*y1+3.0%*z1≤1.0%*3000

0.5%*x2+2.0%*y2+3.0%*z≤2.0%*2000

0.5%*x3+2.0%*y3+3.0%*z3 1.0%*1000

⑤ 生产能力：x1+y1+z1+x2+y2+z2+x3+y3+z3 ≤14000

目标函数为利润，表达式如下：

z = 0.0070*3000 + 0.0060*2000 + 0.0050*1000－0.0004*6000－0.0045*(x1+x2+x3)－0.0035*(y1+y2+y3)－0.0025*(z1+z2+z3)

=35.6000－0.0045*(x1+x2+x3)－0.0035*(y1+y2+y3)－0.0025*(z1+z2+z3)　(万元)

若令c=[0.0045,0.0045,0.0045,0.0035,0.0035,0.0035,0.0025,0.0025,0.0025],

x=[x1,x2,x3,y1,y2,y3,z1,z2,z3],

则目标函数可简写成 Z=35.6000－c*x

求 Z 的最大值即求 min c*x

在命令区中运行如下程序：

```
c=[0.0045,0.0045,0.0045,0.0035,0.0035,0.0035,0.0025,0.0025,0.0025];
a1=[eye(3),eye(3),eye(3)];
a2=[ones(1,3),zeros(1,6);zeros(1,3),ones(1,3),zeros(1,3);zeros(1,6),ones(1,3)];
a3=[－0.1200*eye(3),－0.0600*eye(3),－0.0800*eye(3);0.0050*eye(3),0.0800*eye(3),0.0800*eye(3)];
A=[a1;a2;a3;ones(1,9)];
b=[3000,2000,1000,5000,5000,5000,－300,－160,－60,30,40,10,14000]';
v1=zeros(1,9);
x=lp(c,A,b,v1,[],[],3);
x',z=35.6000－c*x,
```

运行结果，如图26-1所示。

```
ans =
  1.0e+003 *
  Columns 1 through 7
    2.4000    0.8000    0.8000         0         0         0    0.6000
```

Columns 8 through 9

1.2000 0.2000

z =

12.6000

由计算结果得到结论为：每日购进原油 A 4000 桶，原油 C 2000 桶，利润为 12.6000 万元。

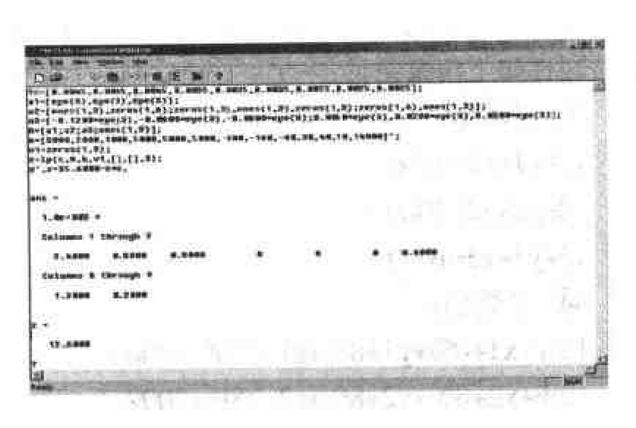


图 26-1 原油计算运行结果图

当投入广告费时，一天向一种汽油投入一元广告费，可使这种汽油日销量增加10桶。设对甲、乙、丙三种汽油分别投入广告费w1，w2，w3元，则它们的日需求量分别增长为 3000+10*w1，2000+10*w2，1000+10*w3。

因此约束条件变为：

① 需求量：

x1+y1+z1=3000+10*w1

x2+y2+z2=2000+10*w2

x3+y3+z3=1000+10*w3

② 买入量：

x1+x2+x3≤5000

y1+y2+y3≤5000

z1+z2+x3≤5000

② 辛烷值：

12%*x1+6%*y1+8%*z1≥10%*(3000+10*w1)

12%*x2+6%*y2+8%*z2≥8%*(2000+10*w2)

12%*x3+6%*y3+8%*z3≥6%*(1000+10*w3)

④ 含量：

0.5%*x1+2.0%*y1+3.0%*z1≤1.0%*(3000+10*w1)

0.5%*x2+2.0%*y2+3.0%*x2≤2.0%*(2000+10*w2)

0.5%*x3+2.0%*y3+3.0%*z3≤1.0%*(1000+10*w3)

⑤ 生产能力：x1+y1+z1+x2+y2+z2+x3+y3+z3 ≤14000

目标函数利润表达式变为：

z = 0.0070 * (3000 + 10 * w1) + 0.0060 * (2000 + 10 * w2) + 0.0050 * (1000 + 10 * w3) − 0.0004*(6000+10*w1+10*w2+10*w3) − 0.0001*(w1+w2+w3) − 0.0045*(x1+x2+x3) − 0.0035*(y1+y2+y3) − 0.0025*(z1+z2+z3)

=35.6000+0.0659*w1+0.0559*w2+0.0459*w3 − 0.0045*(x1+x2+z3) − 0.0035*(y1+y2+y3) − 0.0025*(z1+z2+z3) (万元)

若令 c=[0.0045,0.0045,0.0045,0.0035,0.0035,0.0035,0.0025,0.0025,0.0025, −0.0659, −0.0559, −0.0459],

x=[x1,x2,x3,y1,y2,y3,z1,z2,z3,w1,w2,w3],

则目标函数可简写成 Z=35.6000−c*x

求 Z 的最大值即求 min c*x

在命令区中运行如下程序：

c=[0.0045,0.0045,0.0045,0.0035,0.0035,0.0035,0.0025,0.0025,0.0025, −0.0659, −0.0559, −

```
0.0459];
    a1=[eye(3),eye(3),eye(3),－10*eye(3)];
    a2=[ones(1,3),zeros(1,9);zeros(1,3),ones(1,3),zeros(1,6);zeros(1,6),ones(1,3),zeros(1,3)];
    a3=[－0.1200*eye(3),－0.0600*eye(3),－0.0800*eye(3),diag([1.0000,0.8000,0.6000]);
    0.0050*eye(3),0.0200*eye(3),0.0300*eye(3),－diag([0.1000,0.2000,0.1000])];
    A=[a1;a2;a3;ones(1,9),zeros(1,3)];
    b=[3000,2000,1000,5000,5000,5000,－300,－160,－60,30,40,10,14000]';
    v1=zeros(1,12);
    x=lp(c,A,b,v1,[],[],3);
    x',z=35.6000－c*x,
```

运行结果为：

```
ans=1.0e+003 *
  Columns 1 through 7
    2.0000    2.3333    0.6667    1.0000    3.6667    0.3333    －0.0000
  Columns 8 through 12
    3.5000    －0.0000         0    0.7500         0
z=28.7750
```

结论：每日购进原油 A 5000 桶，原油 B 5000 桶，原油 C 3500 桶，同时对乙种汽油投入广告费 750 元，利润为 28.7750 万元。将结果列表如下：

	甲种汽油（桶/天）			乙种汽油（桶/天）			丙种汽油（桶/天）			利润(万元)
	原油A（桶/天）	原油B（桶/天）	原油C（桶/天）	原油A（桶/天）	原油B（桶/天）	原油C（桶/天）	原油A（桶/天）	原油B（桶/天）	原油C（桶/天）	
不作广告	2400	0	600	800	0	1200	800	0	200	12.60
合计（桶）	3000			2000			1000			6000
对甲做广告	2000	1000	0	2333	3667	3500	667	333	0	28.7750
合计（桶）	3000			9500（作广告需求增加）			1000			13500

（2）一基金管理人的工作是，每天将现有的美元、英镑、马克、日元四种货币按当天汇率相互兑换，使在镑足需要的条件下，按美元计算的价值最高。设某天的汇率、现有货币和当天需求如下表：

	美元	英镑	马克	日元	现有量（$\times10^8$）	需求量（$\times10^8$）
美元	1	.58928	1.743	138.3	8	6
英镑	1.697	1	2.9579	234.7	1	3
马克	.57372	.33808	1	79.346	8	1
日元	.007233	.00426	.0126	1	0	10

问该天基金管理人员如何操作（“按美元计算的价值”指兑入、兑出汇率的平均值，如 1 英镑相当于(1.679+(1/.58928))/2=1.696993 美元）。

我们设8亿美元中有m1，m2，m3亿分别兑换成英镑、马克、日元（剩余8－(m1+m2+m3)亿），1亿英镑中有y1，y2，y3亿分别兑换成美元、马克、日元（剩余1－(y1+y2+y3)亿），8亿马克中有k1，k2，k3亿分别兑换成美元、英镑、日元（剩余8－(k1+k2+k3)亿），则有下列约束余件：

8－(m1+m2+m3)≥0

1－(y1+y2+y3)≥0

8－(k1+k2+k3)≥0

兑换以后，需求量应满足下列约束条件：

美元数量：Z1=8－(m1+m2+m3)+1.697*y1+0.57372*k1≥6

英镑数量：Z2=1－(y1+y2+y3)+0.58928*m1+0.33808*k2≥3

马克数量：Z3=8－(k1+k2+k3)+1.743*m2+2.9579*y2≥1

日元数量：Z4=138.3*m3+234.7*y3+79.346*k3≥10

目标函数为“按美元计算的价值”，表达式如下：

Z=Z1+Z2*(1.679+1/0.58928)/2+Z3*(0.57372+1/1.743)/2+Z4*(0.007233+1/138.3)/2

若令 d1=(1.697+1/0.58928)/2,d2=(0.57372+1/1.743)/2,d3=(0.007233+1/138.3)/2;

c=[1－0.58928*d1,1－1.743*d2,1－138.3*d3,d1－1.697,d1－2.9579*d2,
    d1－234.7*d3,d2－0.57372,d2－0.33808*d1,d2－79.346*d3];

x=[m1,m2,m3,y1,y2,y3,k1,k2,k3]'

则目标函数可简写成 Z=8+d1+8*d2－c*x

求 Z 的最大值即求 min c*x

在命令区中运行如下程序：

```
a1=[ones(1,3),zeros(1,6);zeros(1,3),ones(1,3),zeros(1,3);zeros(1,6),ones(1,3)];
a2=[1,1,1,－1.697,0,0,－0.57372,0,0;－0.58928,0,0,1,1,1,0,－0.33808,0;
0,－1.743,0,0,－2.9579,0,1,1,1;0,0,－138.3,0,0,－234.7,0,0,－79.346];
A=[a1;a2];
b=[8,1,8,2,－2,7,－10]';
d1=(1.697+1/0.58928)/2;
d2=(0.57372+1/1.743)/2;
d3=(0.007233+1/138.3)/2;
c=[1－0.58928*d1,1－1.743*d2,1－138.3*d3,d1－1.697,d1－2.9579*d2,d1－234.7*d3,d2－
0.57372,d2－0.33808*d1,d2－79.346*d3];
v1=zeros(1,9);
x=lp(c,A,b,v1);
x',z=8+d1+8*d2－c*x,
```

运行结果，如图 26-2 所示。

```
ans =
  Columns 1 through 7
    5.0910         0         0   －0.0000    0.0000    1.0000    5.3876
```

```
  Columns 8 through 9
        0    1.6124
z =
   14.2872
```

接着运行以下程序，计算兑换后美元、英镑、马克和日元的数量：

```
Z1=8－(x(1)+x(2)+x(3))+1.697*x(4)+0.57372*x(7);
Z2=1－(x(4)+x(5)+x(6))+0.58928*x(1)+0.33808*x(8);
Z3=8－(x(7)+x(8)+x(9))+1.743*x(2)+2.9579*x(5);
Z4=138.3*x(3)+234.7*x(6)+79.346*x(9);
Z1,Z2,Z3,Z4
```

运行结果为：

```
Z1=6.00000000000000
Z2=3
Z3=1.00000000000000
Z4=3.626396222008447e+002
```



图 26-2 资金操作运行结果图

结论：该天基金管理人员应将 5.09095846 亿美元兑换成英镑，将 1 亿英镑兑换成日元，将 5.38757313 亿马克兑换成美元，将 1.61242687 亿马克兑换成日元，这样兑换后，按美元计算的价值最大为 14.28724870 亿美元。兑换后，美元有 6 亿，英镑有 3 亿，马克有 1 亿，日元有 362.63962220 亿，均能满足需求。

【练习小结】

本练习主要介绍了优化计算在实际生活中的运用，原油的生产和作广告对销售的影响，在资金操作中如何取得最大利润，这些都是现实生活中的实际问题，而优化计算给出了令人满意的答案。

【思考题】

将本练习中的例题在 MATLAB 中实现。

练习 27 数 值 积 分

知识背景

积分运算是工程数学中的重要的一环，对于一般的表达式以及较简单的被积分式，我们可以将其运算而得出解析解并进而计算出解析值。然而在绝大多数情况下，当被积分式较复杂、被积函数“积不出来”或者积分式和积分区间是离散的情况时，我们只有通过一定的方法来求出其近似解或称数值解。用数值方法求一个未知表达式的函数 f(x)在区间(a,b)上的定积分 $S=\int_a^b f(x)$ 的基本思路是：

$$S=\int_a^b f(x)dx=\lim_{n\to\infty} I_n, I_n=\sum_{k=1}^{n} f(\xi_k)\frac{b-a}{n}$$

当 n 充分大时，就是 S 的数值积分值。而各种不同的数值积分方法的区别就在于选择不同的简单函数代替 f(x)。

计算数值积分的梯形公式和辛普森求积公式，前者主要用分段线性插值函数作为 f(x)的近似，后者则选择分段二次插值函数代替 f(x)。其中梯形公式是二阶收敛的，而辛普森公式则是四阶收敛的。此外，还有高斯（Guass）求积公式和蒙特卡罗求积公式，前者通过提高积分函数的代数精度来提高积分数值的准确性；而后者则是将被积函数的被积区间变换到（0，1），用被计的方法来在（0，1）区间内取随机数，从而实现积分计算。然而，由于计算机实际上并不可能产生真正的随机数（只被称之为伪随机被），只可能产生一定精度的随机数，故蒙特卡罗方法精度较低。

主要内容

【本练习考查知识点】

本练习将介绍一些积分运算中常见的函数，要求读者能够初步了解梯形公式和辛普森公式计算数值积分，并掌握基本的积分运算函数。

练习过程

（1）我们首先来熟悉一下各种积分算法，我们用不同的方法来计算以下积分：

$$\int_0^{\frac{\pi}{2}} \cos x dx$$

① 先用矩形公式和梯形公式，这里用到两个函数 sum(x)和 cumsum(x),其中 sum 是求和公式，输入数组 x，输出为 x 的和；而 cusum 输入数组 x，输出数组为数组 x 的依次累加和。将$(0,\frac{\pi}{2})$ 15 等分，步长为$\frac{\pi}{30}$，程序如下：

```
l=pi/30;
x=0:l:pi/2
y=cos(x);
z1=sum(y(1:15))*l;
z2=sum(y(2:16))*l;
z=cumsum(y);
z3=z(15)*l;
z4=(z(16)-z(1))*l;
z5=trapz(y)*l;
z1,z2,z3,z4,z5
```

这里显示运行结果如图 27-1 所示。

运行结果显示，z1=z3=1.0514，z2=z4=0.9467，z5=0.9991，与解析解比较接近。

② 辛普森公式计算，可以采用函数 quad，它采用迭代算法。也可以用 quad8，它采用自适应 Newton Cote 8 panel 法则。运行程序为：

```
z6=quad('cos',0,pi/2)
```

运行结果如图 27-2 所示。

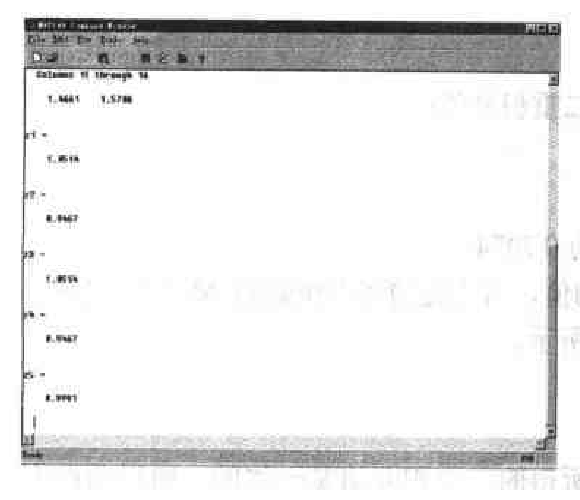


图 27-1　梯形和矩形公式运行结果图

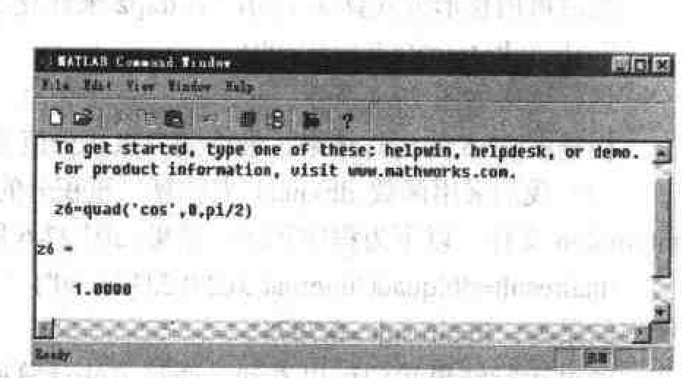


图 27-2　辛普森公式运行结果图

z6=1.0000 结果很精确，说明 quad 函数能够比较准准地计算积分值。

③ 准特卡罗统计方法计算积分值的程序如下：

```
n=8000;
x=rand(1,n);
y=cos(x.*pi/2);
z7=sum(y)*pi/2/n
```

运行结果如图 27-3 所示。

由于算法是通过取准机数来进行运算的，所以每一次运行的结果都不相等。

（2）以上练习是关于一准积分的，以下我们来练习二够积分的运算。二维积分的运算可以有两种方法。其一是通过准层积分的方式，即先用辛普森算法的 quad 函数计算内层积分值，再用梯形公式计算外层的积分值；第二种方法是用公式 dblquad 直接对被积分式进行计算。当求二维积分：

$\int_{y\min}^{y\max}\int_{x\min}^{x\max} f(x,y)dxdy$ 的值时，可以调用如下格式的函数：

```
S=dblquad('double',xmin,xmax,ymin,ymax,TOL,'METHOD')
```

其中，double 为存放被积分函数表达式的文件名，xmax、xmin 是内层积分的上下限，ymax,ymin 是外层积分的上下限，而 TOL 为可选项。TOL 为误差限，METHOD 为积分方法，缺省值为 quad 法。如计算二重积分：

$$\int_0^2\int_0^2 e^{-x^3-y^3}dxdy$$

采用第一种方法，首先创建一个 M 文件：internal.m，如图 27-4 所示：

```
function y=internal(x,y)
y=exp(-x.^3-y.^3);
```

然后用函数 quad 计算固定 x 而在 y 方向上的一些积分值：

```
x=linspace(0,2,40);
for i=1:40
innerresult(i)=quad('internal',0,2,[],[],x(i));
end
```

然后再用梯形公式算法中的函数 trapz 来计算二重积分值：

```
finalresult=trapz(x,innerresult)
```

运行结果如图 27-4 所示。

根据图 27-5 的运行结果，最终的二重积分值为 0.7974。

（3）我们采用函数 dblquad 来计算二重积分的值，所创建的包含函数的 M 文件仍然用 internal.m 文件，以下为程序和运行结果如图 27-6 所示。

```
finalresult=dblquad('internal',0,2,0,2,[],'quad')
```

所得结果为：finalresult =0.7974。

通过运行结果我们可以看到，两种方法计算所得的二重积分值是一致的，所以两种方法都可以计算二重积分值，只不过第二种方法比较简便和快捷。

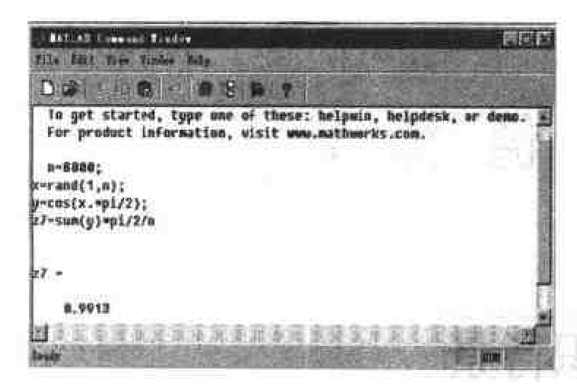


图 27-3　蒙特卡罗法运行结果图

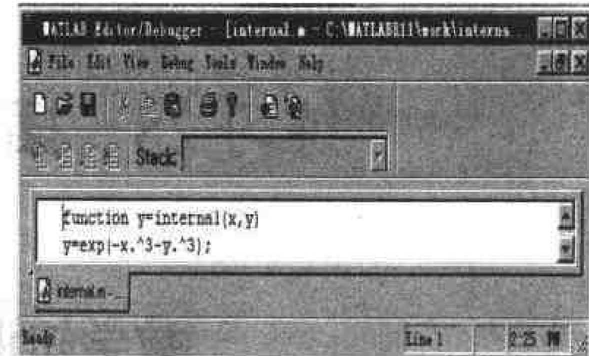


图 27-4　internal.m 文件图

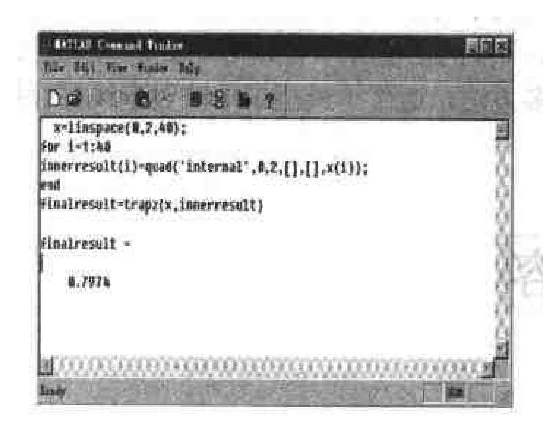


图 27-5　分步计算法运行结果图

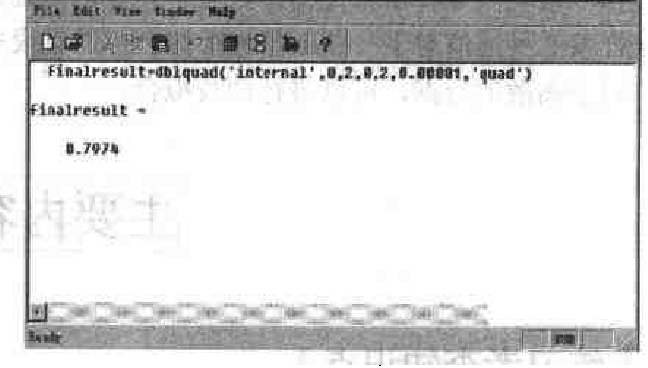


图 27-6　函数公式法运行结果图

【练习小结】

本练习主要介绍了一重和二重数值积分的一些主要算法和相关函数，并用一些例子程序来向读者介绍。希望读者通过本练习能够了解一重数值积分计算中矩形、梯形、辛普森算法和蒙特卡罗统计算法计算数值积分的原理，并能熟练掌握各相关函数计算数值积分。二重积分计算主要的两种方法，希望读者能够用函数 dblquad 方法计算二重数值积分值。

【思考题】

1. 请用文中介绍的方法分别计算积分值：：$\int_0^1 \sin x dx$

2. 请用梯形法、辛普森法和蒙特卡罗法分别计算积分值：$\int_0^1 \sqrt{x^2+x+1}dx$

3. 请分别用文中介绍的方法计算二重积分值：$\int_0^1\int_0^2 (x^2+y^2+xy+2x+y+1)dxdy$

练习 28 插 值 拟 合

数学知识背景

插值在机械加工等工程技术和数据处理等科学研究中有很多直接的运用，而且插值又是数值积分数值微分和常微分方程数值解计算的基础。主要的插值方法有拉格朗日、分段线性、三次样条三种插值对于一些数据进行拟合可以有很多方法，通常用最小二乘法对直线进行拟合。对于离散的数据，可以进行曲线拟合。

主要内容

【本练习考查知识点】

本练习介绍 MATLAB 计算拉格朗日、分段线性、三次样条三种插值的方法，改变节点的数目，对三种插值结果进行初步分析。通过实例学习如何用插值方法与拟合方法解决实际问题。

练习过程

（1）函数 $y=\exp(-x^2)$ $(-2\leqslant x\leqslant 2)$，在 n 个节点上（n 不要太大，如 5~11）用拉格朗日、分段线性、三次样条三种插值方法，计算 m 个插值点的函数值（m 要适中，如 50~100）。通过数值和图形输出，将三种插值结果与精确值进行比较。适当增加 n，再作比输，由此作初步分析。

编制计算拉基朗日插值的名为 lagr1 的 M 文件：

```
function y=lagr1(x0,y0,x)
n=length(x0);m=length(x);
for i=1:m
    z=x(i);

s=0.0;
for k=1:n
```

```
        p=1.0;
        for j=1:n
            if j~=k
                p=p*(z-x0(j))/(x0(k) -x0(j));
            end
        end
        s=p*y0(k)+s;
    end
    y(i)=s;
    end
```

在命令区中运行如下程序：

```
n=5;m=61;x=-2:4/(m-1):2;
y=exp(-x.^2);
z=0*x;
x0=-2:4/(n-1):2;
y0=exp(-x0.^2);
y1=lagr1(x0,y0,x);
y2=interp1(x0,y0,x);
y3=interp1(x0,y0,x,'spline');
[x'y'y1'y2'y3']
plot(x,z,'r',x,y,'k:',x,y1,'r',x,y2,'m',x,y3,'b')
gtext('Lagr.'),gtext('Piece.-linear.'),gtext('Spline'),gtext('y=exp(-x^2)')
```

运行后，所得程序结果如图 28-1 所示，得到各节点和插值点的值如下：（因篇幅问题，此处没有全部列出）

x	y	y1	y2	y3
0	1.0000	1.0000	1.0000	1.0000
0.0667	0.9956	0.9966	0.9579	0.9956
0.1333	0.9824	0.9865	0.9157	0.9828
0.2000	0.9608	0.9698	0.8736	0.9623
0.2667	0.9314	0.9465	0.8314	0.9349
……				
1.0000	0.3679	0.3679	0.3679	0.3679
1.0667	0.3205	0.3010	0.3446	0.3101
1.2000	0.2369	0.1714	0.2980	0.2011
1.2667	0.2010	0.1108	0.2747	0.1512
……				
1.8667	0.0307	−0.0867	0.0649	−0.0348
1.9333	0.0238	−0.0437	0.0416	−0.0137
2.0000	0.0183	0.0183	0.0183	0.0183

将三种插值结果 y1，y2，y3 与精确值 y 相比较，发现它们在节点处相等，而在插值点处 y3（Spline 插值）的结果最好。增加节点以后，三种插值方法得到的插值结果与精确值相比都很贴近。可以初步得出结论，增加节点个数，可以提高插值结果的准确性。

（2）用给定的多项式，如 $y=x^3-6x^2+5x-3$，产生一组数据（$x_i,y_i,i=1,2,\cdots,n$），再在 y_i 上添加随机干扰（可用 rand 产生（0，1）均匀分布随机数，或用 randn 产生 N（0，1）分布随机数），然后用 x_i 和添加了随机干扰的 y_i 作 3 次多项式拟合，与原系数比较。如果作 2 或 4 次多项式拟合，结果如何？

在命令区中运行如下程序：

```
n=11;
x=0:10/(n-1):10;
y=x.^3-6*x.^2+5*x-3;
z=0*x;
y1=y+rand(1,11);
a1=polyfit(x,y1,3);
z1=polyval(a1,x);a2=polyfit(x,y1,2);
z2=polyval(a2,x);
a3=polyfit(x,y1,4);
z3=polyval(a3,x);
plot(x,z,'r',x,y,'k:',x,z1,'m',x,z2,'r',x,z3,'b')
```

运行后，得到结果如图如图 28-2 所示（近似水平的红色线表示四次多项式，红色曲线表示二次多项式，而蓝色曲线表示三次多项式）。红色比较拟合后多项式和原式的系数，发现三次多项式系数与原系数比较接近，四次多项式的四次项系数很小。作图后，发现二次多项式与原函数的差别比较大，而三次多项式和四次多项式符合得比较好。

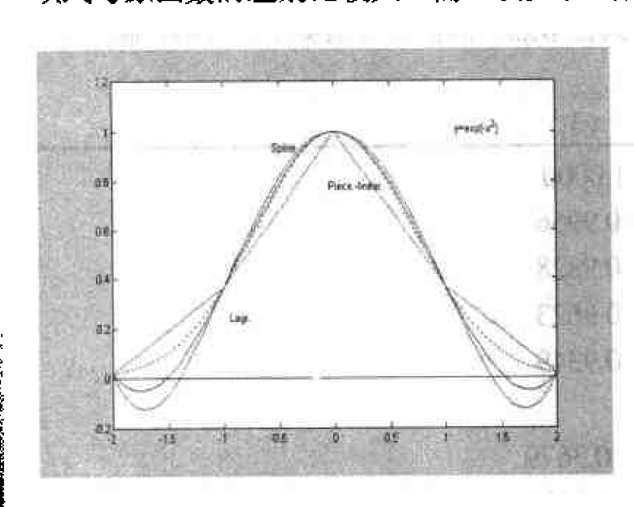


图 28-1 三种插值运行图

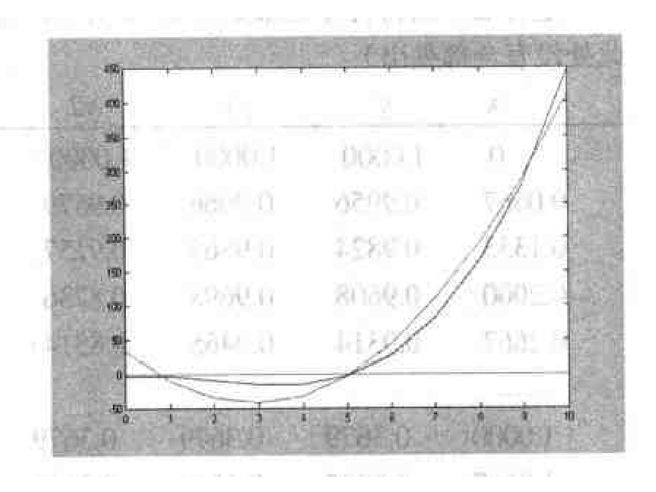


图 28-2 多项式拟合曲线

（3）弹簧在力 F 的作用下伸长 x，一定范围内服从胡克定律：F 与 x 成正比，即 F=kx，k 为弹性系数。现在得到下面一组 x，F 数据，并在（x，F）坐标下作图。可以看出，当 F 大到一定数值后，就不服从这个定律了。试由数据确定 k，并给出不服从胡克定律时的近似公式。

X	1	2	4	7	9	12	13	15	17
F	1.5	3.9	6.6	11.7	15.6	18.8	19.6	20.6	21.1

弹簧在力F下的伸长x

25
20
15
F
10
5
0
0 2 4 6 8 10 12 14 16 18
X

根据图中点的位置特点，把前 5 个点拟合成直线，把后 5 个点拟合成二次曲线。

运行如下程序：

```
x=[1,2,4,7,9,12,13,15,17];
f=[1.5,3.9,6.6,11.7,15.6,18.8,19.6,20.6,21.1];
a1=polyfit(x(1:5),f(1:5),1);
x0=9－0.001; %为保证连接光滑，二次曲线在x=9这一点斜率应趋近于直线的斜率。
y0=a1(1)*x0+a1(2);
x2=[x0,x(5:9)];f2=[y0,f(5:9)];
a2=polyfit(x2,f2,2);
y1=polyval(a1,x(1:5));
y2=polyval(a2,x(5:9));
a1,a2
plot(x(1:5),y1,'r',x(5:9),y2,'b')
```

运行后，得到的曲线如图 28-3 所示。

弹簧在不服从胡克定律时的近似公式为 $F=-0.0803x^2+2.7865x-3.0822$。

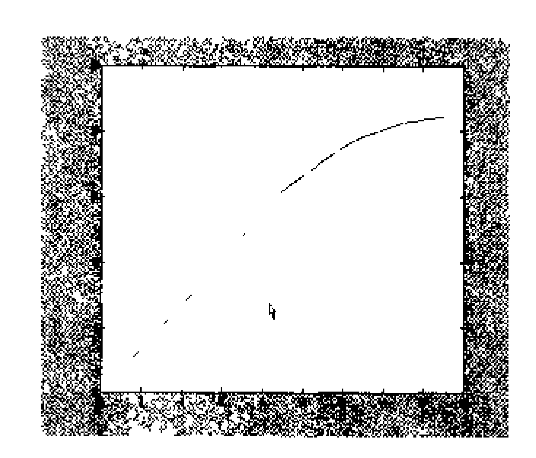


图 28-3　弹簧弹力于伸长关系图

【练习小结】

本练习主要是介绍了插值拟合计算，插值方式主要有拉格朗日、分段线性、三次样条三种插值的方法，而数据拟合主要有线形和非线形最小二乘拟合方法。

【思考题】

1．已知：x=[0,3,5,7,9,11,12,13,14,15]，y=[0,1.2,1.7,2.0,2.1,2.0,1.8,1.2,1.0,1.6]，假设需要得到 x 坐标每改变 0.1 时的 y 坐标，试求出所需数据，并画出曲线。

2．已知模型：$t = an^2 + bn$

测得对应的数据(t,n)如下：

t=[0,20,40,60,80,100,120,140,160,183.5]；

x=[0,1153,2045,2800,3466,4068,4621,5135,5619,6152]；

试求 a 和 b 的值。

练习 29　回 归 分 析

数学知识背景

回归分析是对拟合问题进行的统计分析，数据拟合的好坏直接影响对问题研究的正确与否，也直接关系到模型建立正确与否。通过回归分析可以对数据拟合的好坏进行分析，从而为建立模型和研究问题提供有力的支持。

主要内容

【本练习考查知识点】

本练习主要介绍回归分析的基本思想和在 MATLAB 中的实现方法，讲述如何在实际问题中运用回归分析来解决问题。

练习过程

（1）用切削机床加工时，为实时地调整机床需测定刀具的磨损速度，现每隔一小时测量刀具的厚度得到以下数据，试建立刀具厚度关于切削时间的回归模型，对模型和回归系数进行检验，预测 15 小时后刀具的厚度。

时间(h)	0	1	2	3	4	5	6	7	8	9	10
刀具厚度(cm)	30.6	29.1	28.4	28.1	28.0	27.7	27.5	27.2	27.0	26.8	26.5

先作出各个点的位置。在命令区运行程序：

```
x=[0,1,2,3,4,5,6,7,8,9,10]';
y=[30.6,29.1,28.4,28.1,28.0,27.7,27.5,27.2,27.0,26.8,26.5]';
plot(x,y,'r+')
```

运行后得到如下图结果。从图 29-1 中可见，y 与 x 大致符合线性关系。

接着运行如下程序：

```
X=[ones(11,1),x];
```

```
[b,bint,r,rint,stats]=regress(y,X);
b,bint,stats,rcoplot(r,rint)
```

运行结果为：

```
b = 29.5455
    −0.3291
bint = 28.9769    30.1140
       −0.4252   −0.2330
stats =0.8696    60.0018      0.0000
```

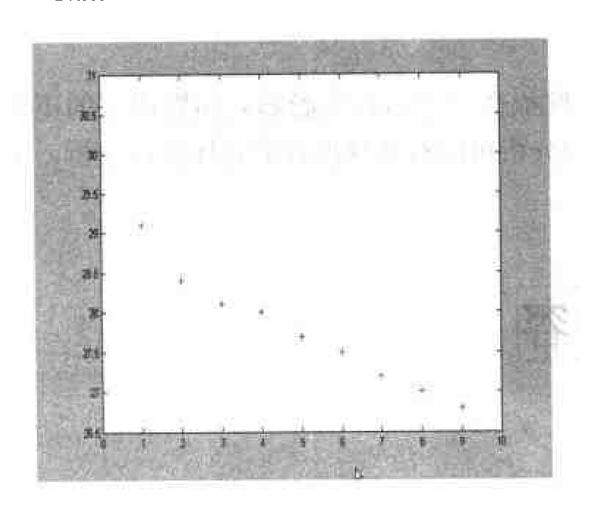


图 29-1　磨损实验图

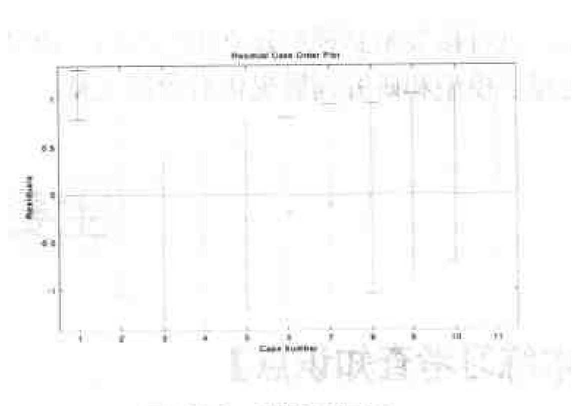


图 29-2　残差分布图

由以上结果可见，相关系数 R^2=0.8696，则 R=0.9325>0.9，$p<10^{-4}$，因此，此回归模型是成立的。以此预测 15 小时后刀具的厚度：

y=b(1)+b(2)*15

结果为：y =24.6091，即 15 小时后刀具的厚度为 24.6091cm。

观察图中的残差分布，除第一个点以外，其余残差的置信区间均包含零点，第一个点应视为异常点，将其剔除后重新计算，可得：

```
b = 29.0533        bint = 28.8334    29.2732        stats =0.9726   283.5599    0.0000
   −0.2588                −0.2942   −0.2233
```

可见，回归系数变化不大，但其置信区间明显缩小，R^2 和 F 都明显变大。因此，用修改后的结果预测应该更合适。计算 y=b(1)+b(2)*15，结果为：y =25.1713，即 15 小时后刀具的厚度为 25.1713cm。

（2）得到某商品的需求量与消费者的平均收入、商品价格的统计数据如下，建立回归模型并进行检验，预测平均收入为 1000、价格为 6 时的商品需求量。

需求量	100	75	80	70	50	65	90	100	110	60
收入	1000	600	1200	500	300	400	1300	1100	1300	300
价格	5	7	6	6	8	7	5	4	3	9

先作图观察需求量与收入、价格的大致关系。在命令区运行程序：

```
x1=[1000,600,1200,500,300,400,1300,1100,1300,300]';
```

```
x2=[5,7,6,6,8,7,5,4,3,9]';
y=[100,75,80,70,50,65,90,100,110,60]';
plot(x1,y,'r+')
plot(x2,y,'r+')
```

分别得到图 29-3 和图 29-4，从两个图中可见，y 分别与 x1，x2 大致符合线性关系。

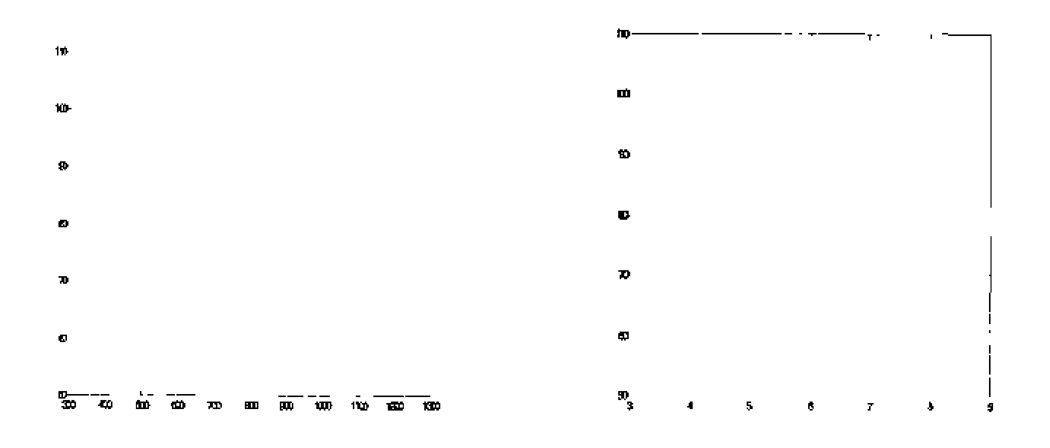


图 29-3　需求量与价格关系图　　　　图 29-4　收入与价格关系图

接着运行如下程序：

```
x=[x1,x2];
X=[ones(10,1),x];
[b,bint,r,rint,stats]=regress(y,X);
b,bint,stats,rcoplot(r,rint)
```

运行结果为：

```
b = 111.6918
     0.0143
    −7.1882
bint = 56.0503    167.3334
       −0.0120     0.0406
       −13.2306   −1.1458
stats =0.8944     29.6533      0.0004
```

并得到图 29-5。

由以上结果可见，相关系数 R^2=0.8944，则 R=0.9457>0.9，p=0.0004<0.05，观察图中的残差分布，各个点的置信区间均包含零点。因此，此回归模型 y=111.6981+0.0143∗x1－7.1882∗x2 是成立的。

下面再用 rstool 命令来预测平均收入为 1000、价格为 6 时的商品需求量：

```
rstool(x,y)
```

得到结果如图 29-6 所示。

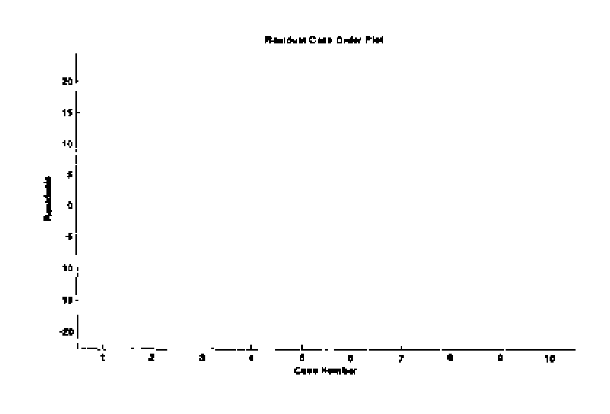


图 29-5 残差分布图

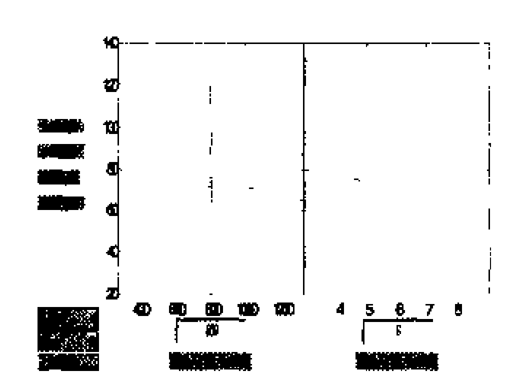


图 29-6 收入与需求量交互式画面

预测平均收入为 1000、价格为 6 时的商品需求量为 83，向工作区输出回归系数和剩余标准差：

```
beta,rmse
beta = 111.6918
         0.0143
        -7.1882
rmse = 7.2133
```

通过改变模型，分别用纯二次、交叉二次和完全二次进行分析，发现用完全二次模模的剩余标准差最小。此时：

```
beta =  -106.6095          rmse = 4.4179
           0.3261                 21.2990
          -0.0200
          -0.0001
          -0.7609
```

预测平均收入为 1000、价格为 6 时的商品需求量为 89。

（3）一矿脉有 13 个相邻样本点，人为地设定一原点，现测得各样本点对原点的距离 x，与该样本点处某种金属含量 y 的一组数据如下，画出散点图观察二者的关系，试建立合适的回归模型，如二次曲线、双曲线、对数曲线等。

x	2	3	4	5	7	8	10
y	106.42	109.20	109.58	109.50	110.00	109.93	110.49
x	11	14	15	15	18	19	
y	110.59	110.60	110.90	110.76	111.00	111.20	

先画出散点图。在命令区运行程序：

```
x=[2,3,4,5,7,8,10,11,14,15,15,18,19]';
y=[106.42,109.20,109.58,109.50,110.00,109.93,110.49,110.59,110.60,110.90,110.76,111.00,11
1.20]';
plot(x,y,'r+')
```

可得散点分布图如图 29-7 所示。

建立不同的模型进行验证：

① **y=a+bx**，在命令区运行以下程序：

```
X=[ones(13,1),x];
[b,bint,r,rint,stats]=regress(y,X);
b,bint,stats,rcoplot(r,rint)
```

运行结果为：

```
b = 108.2581
    0.1742
bint = 107.2794   109.2367
         0.0891     0.2593
stats = 0.6484      20.2866       0.0009
```

运行结果如图 29-8 所示。

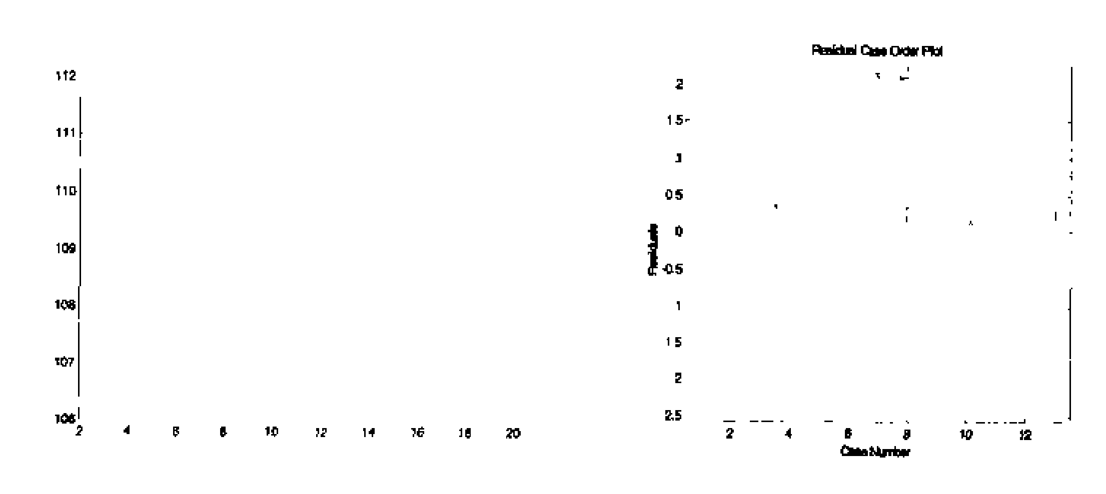


图 29-7　散点部分图　　　　图 29-8　残差分布图

由以上结果可见，相关系数 R^2=0.6484，则 R=0.8052>0.8，p=0.0009<0.05，模型 y=108.2581+0.1742*x 基本成立。计算剩余标准差：

```
Y=b(1)+b(2)*x;
s=sqrt(sum((y－Y).^2)./11)
```

结果为：s =**0.7723**

观察图 29-8 的残差分布，除第一个点以外，其余残差的置信区间均包含零点，第一个点应视为异常点，将其剔除后重新计算，可得：

```
b = 109.0668
    0.1159
bint = 108.8264   109.3072
         0.0958     0.1360
stats =0.9428     164.8060      0.0000
```

剩余标准差为：s =**0.8818**，置信区间明显缩小，R^2 和 F 明显变大，但剩余标准差也会变大。如果用来作预测，则最好用这个模型，即 y=109.0668+0.1159*x。

② **y=a+b/x**，用非线性最小二乘法拟合。编写 f118.m 文件如下：

```
function f=f118(b)
x=[2,3,4,5,7,8,10,11,14,15,15,18,19];
y=[106.42,109.20,109.58,109.50,110.00,109.93,110.49,110.59,110.60,110.90,110.76,111.00,111.20];
f=y－(b(1)+b(2)./x);
```

在命令工作区输入命令：

```
b0=[50,10];
b=leastsq('f118',b0)
```

运行结果为：

b =111.4405 －9.0300

即模型为 y=111.4405－9.0300/x。

计算剩余标准差：

```
s=sqrt(sum(f118(b).*f118(b))./11)
```

结果为：s =**0.3440**

③ **y=a+bx+cx^2**，运行如下程序：

```
x=[2,3,4,5,7,8,10,11,14,15,15,18,19];
y=[106.42,109.20,109.58,109.50,110.00,109.93,110.49,110.59,110.60,110.90,110.76,111.00,111.20];
[p,S]=polyfit(x,y,2);
p
```

运行结果为：

p =－0.0170 0.5271 106.9522

即模型为 y=－0.0170x2+0.5271x+106.9522。

计算剩余标准差：

```
[Y,delta]=polyconf(p,x,S);
s=sqrt(sum((y－Y).^2)./11)
```

结果为：s =**0.6166**

④ **y=a+bx$^{1/2}$**，修改 f118.m 文件为：

```
f=y－(b(1)+b(2).*sqrt(x));
```

运行如下程序：

```
b0=[100,1];
b=leastsq('f118',b0)
```

运行结果为：

b =106.6901 1.0957

即模型为 y=106.6901+1.0957x$^{1/2}$。计算剩余标准差：

```
s=sqrt(sum(f118(b).*f118(b))./11)
```

结果为：s =**0.6693**

⑤ y=a+blnx

修改 f118.m 文件为：

```
f=y-(b(1)+b(2).*log(x));
```

运行如下程序：

```
b0=[100,1];
b=leastsq('f118',b0)
```

运行结果为：

```
b =106.7113    1.5663
```

即模型为 y=106.7113+1.5663lnx。

计算剩余标准差：

```
s=sqrt(sum(f118(b).*f118(b))./11)
```

结果为：s = 0.5494

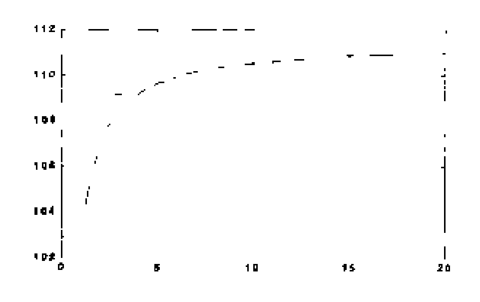


图 29-9　散点与拟合后曲线图

比较以上 5 种模型，发现双曲线的剩余标准差最小，和散点符合较好，作图如图 29-9 所示。

【练习小结】

本练习通过几个数据拟合的回归分析来判断曲线拟合的精度，通过回归分析来判断模型建立是否正确。

【思考题】

复习练习中回归分析的内容，并试着在 MATLAB 中实现本练习的内容。

练习 30　数据统计分析

数学知识背景

数据统计以概率论和数理统计为基础来研究批量数据之间的关系，数理统计研究的对象是受随机因素影响的数据，统计可以从样本推断总体。通过研究概率论与数理统计中各个参数，可以合理地得到结论。关于概率论和数理统计的一些基本概念和公式定理请参考相关的教学参考书和教材。

主要内容

【本练习考查知识点】

本练习主要讲述数据的统计数述和参数估计、假设检验的基本概念与原理，及用 MATLAB 实现的方法，并用这些方法解决实际问题。

练习过程

（1）某厂从一台机床生产的概珠中随机抽取 9 个，测得直径（mm）如下：

14.6，14.7，15.1，14.9，14.8，15.0，15.1，15.2，14.8

设滚珠直径概从正态分布，试自行给出不同的显著性水平，对直径的均值和标准差作区间估计。

在命令窗口中运行如下程序：

```
x=[14.6,14.7,15.1,14.9,14.8,15.0,15.1,15.2,14.8];
alpha=0.01;
[mu sigma muci sigmaci]=normfit(x,alpha)
```

得到结果如下：

```
mu =
     14.9111
```

```
sigma =
    0.2028
muci =
   14.6843
   15.1379
sigmaci =
    0.1224
        0.4946
```

改动分别令 alpha 值为 0.05 和 0.005，得到结果如下：

当 alpha=0.05 时

```
mu =
   14.9111
sigma =
    0.2028
muci =
   14.7553
   15.0670
sigmaci =
    0.1370
0.3884
```

当 alpha=0.005 时

```
mu =
   14.9111
sigma =
    0.2028
muci =
   14.6521
   15.1701
sigmaci =
    0.1176
0.5457
```

比较以上结果可以看出，对于不同的 alpha，得到的均值和标准差是一样的，而置信区间却不同。alpha 值越小，区间内值的可信程度越大，同时置信区间越大。

（2）设（1）中的数据是机床甲产生的，另从机床乙生产的滚珠中抽取 10 个，测得直径（mm）如下：

15.2，15.1，15.4，14.9，15.3，15.0，15.2，14.8，15.7，15.0

设两机床生产的滚珠直径分别为 $\mu 1$，$\mu 2$，试作 $\mu 1=\mu 2$，$\mu 1\leqslant\mu 2$，$\mu 1\geqslant\mu 2$ 三种检验，并检验两机床精度是否一样。

（a）检验 $\mu_1=\mu_2$，运行如下程序：

```
x1=[14.6,14.7,15.1,14.9,14.8,15.0,15.1,15.2,14.8];
```

```
x2=[15.2,15.1,15.4,14.9,15.3,15.0,15.2,14.8,15.7,15.0];
[h,p,ci]=ttest2(x1,x2,0.05,0)
```

结果为：h=1

p=0.0352

ci=－0.4784 －0.0194

因为 h=1，所以拒绝 $\mu_1=\mu_2$ 的假设，故 $\mu_1\neq\mu_2$。

（b）检验 $\mu_1\leq\mu_2$，运行如下程序：

```
[h,p,ci]=ttest2(x1,x2,0.05,1)
```

结果为：h=1

p=0.9824

ci=－0.4784 －0.0194

因为 h=0，所以接受 $\mu_1\leq\mu_2$ 的假设，$\mu_1\leq\mu_2$ 成立。

（c）检验 $\mu_1\geq\mu_2$，运行如下程序：

```
[h,p,ci]=ttest2(x1,x2,0.05,－1)
```

结果为：h=1

p=0.0176

ci=－0.4784 －0.0194

因为 h=1，所以拒绝 $\mu_1\geq\mu_2$ 的假设，$\mu_1\geq\mu_2$ 不成立。

（d）检验精度是否一样，运行如下程序：

```
s1=var(x1);
s2=var(x2);
f=s2/s1
f1=finv(0.975,8,9)
f2=finv(0.025,8,9)
```

结果为：f=1.6865 f1=4.1020 f2=0.2295

因为 f2<f<f1，所以在 0.05 的显著性水平下可以认为方差相同，接受两者精度相同。

所以结论为：甲的均值小于乙的均值；当显著性水平为 0.05 时，甲的精度等于乙的精度。

（3）甲方向乙方成批供货，甲方承诺合格率为 90%，双方商定置信概率为 0.95，现从一批货中抽取 50 件，43 件为合格品，问乙方应否接受这批货物。你能为乙方不接受它出谋划策吗？

样品总体服从 0－1 分布：x=0 表示废品，x=1 表示合格品。若合格品率为 p，则 X 的期望为 μ=p。根据中心极限定理，当样本充分大时，对样本均值 x 有，$z=(x-\mu)/(\sigma/\sqrt{n})$ 近似服从 N(0,1)。因此运行如下程序：

```
x=(43/50－0.9)/sqrt(0.1*0.9/50);
y=norminv(0.95,0,1)
```

得到结果为：

x=－0.9428

y=1.6449

由于－y<x<y，所以乙方应该接受这批产品。要想不接受，可以改变置信概率。

在命令区运行程序：

```
normcdf(z,0,1)
```

得到：ans=0.1729，因此若想不接受这受产品，只需使置信接率为 1－0.1729=0.8271。

【练习小结】

本练习主要是 MATLAB 中所带的函数工具包来对统计数据进行假设检验，在实际运用中，可以通过信设检验来确定数据的准确性和方案的设计。

【思考题】

已知两组数据：

```
x1=[93,85,79,90,78,76,81,85,88,68,92,73,88,84,90,70,69,83,83,85];
x2=[88,89,86,85,87,88,75,93,88,78,86,86,80,89,85,79,78,88,88,90];
```

把两组数据看作独立的两组数信，接接两总体均信是否相等，会得到什么接果。

练习 31 方 差 分 析

数学知识背景

用数理统计分析试验所得结果、鉴别试验中各个因素对试验结果的影响，从而对试验方案进行正确的评价以及对试验方案进行改进。试验结果可以作为评价试验的指标，试验过程中需要考察、可以进行控制的条件，称为因素，因素所处的状态称为水平。处理实际问题可能遇到单因素试验和双（多）因素试验。而处理这些试验结果的统计方法称为单因素方差分析和双因素方差分析。

主要内容

【本练习考查知识点】

本练习介绍方差分析的基本原理，用 MATLAB 的实现方法和用方差分析方法解决实际问题。

练习过程

（1）为比较 5 种品牌合成木板的耐久性，对每个品牌取 4 个样品作摩擦试验，测量磨损量，得以下数据：

品牌 A　2.2　2.1　2.4　2.5；　　　品牌 B　2.2　2.3　2.4　2.6；

品牌 C　2.2　2.0　1.9　2.1；　　　品牌 D　2.4　2.7　2.6　2.7；

品牌 E　2.3　2.5　2.3　2.4。

① 它们的耐久性有无显著差异；

② 计算每种品牌的均值，有选择地作两品牌的比较能得出什么结果。这是一个单因素的方差分析问题，在命令区中运行下面的程序：

```
x=[2.2,2.2,2.2,2.4,2.3;2.1,2.3,2.0,2.7,2.5;        2.4,2.4,1.9,2.6,2.3;2.5,2.6,2.1,2.7,2.4];
p=anova1(x)
```

运行得到结果如图 31-1 所示。

得到结果：p =0.0019。因为 p=0.0019< α =0.05，所以它们的耐久性有非常显著的差异。

③ 在命令区中输入：

```
m=mean(x)
for i=1:4
    for j=i+1:5
[h,p]=ttest2(x(:,i),x(:,j))
end
end
```

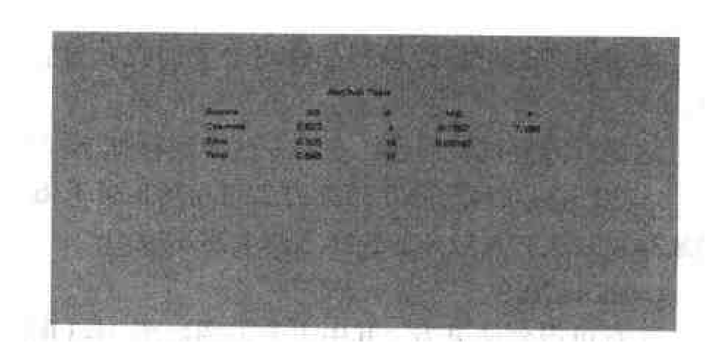


图 31-1　方差表图

运行可得到：

m = 2.3000　　2.3750　　2.0500　　2.6000　　2.3750

品牌两两间比较：

品　牌	A 与 B	A 与 C	A 与 D	A 与 E	B 与 C
h	0	0	1	0	1
p	0.5705	0.0667	0.0408	0.4943	0.0229

品　牌	B 与 D	B 与 E	C 与 D	C 与 E	D 与 E
h	0	0	1	1	1
p	0.0887	1	0.0012	0.0068	0.0388

用 ttest2 对 5 种品牌合成木板的耐久性两两进行假设检验，由结果可看出：C 与 B、D、E 有显著差异，D 与 A、E 也有显著差异，其余两两间差异均不显著。由均值及 Box 图（如图 31-2 所示），有显著差异的也就是均值的差值较大的。

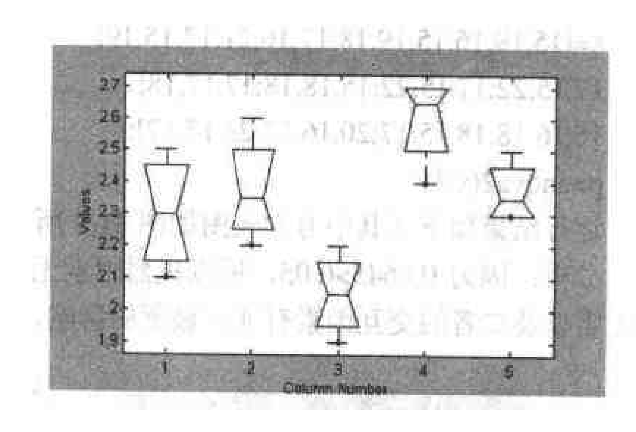


图 31-2　BOX 图

（2）将土质基本相同的一块耕地分成均等的 5 块，每块又分成均等的 4 个小区。在每一块地内，把 4 个品种的小麦分种在 4 个小区，每一小区小麦的播种量相同，测得收获量如下（单位：kg）：

品种 / 地块	A_1	A_2	A_3	A_4	A_5
B_1	32.3	34.0	34.7	36.0	35.5
B_2	33.2	33.6	36.8	34.3	36.1
B_3	30.8	34.4	32.3	35.8	32.8
B_4	29.5	26.2	28.1	28.5	29.4

从这组数据考察地块和品种对小麦的收获量有无显著影响，并在必要时作进一步的比较。

这是双因素试验问题。在命令区中运行如下程序：

```
x=[32.3,34.0,34.7,36.0,35.5;33.2,33.6,36.8,34.3,36.1;
30.8,34.4,32.3,35.8,32.8;29.5,26.2,28.1,28.5,29.4];
p=anova2(x)
```

运行可得到结果为（其中方差表图如图 31-3 所示）：

p =　0.2353　0.0001

结论：p 的两个数值分别是表中因素 A（列）、B（行）的概率，因为 0.2352>0.05，所以因素 A（地块）无显著影响，而因 0.0001<0.01，故因素 B（品种）有非常显著的影响。

（3）下表记录了三位工人分别在四台机器上三天的日产量，分析工人之间的差异是否显著，机器之间的差异是否显著，二者的交互作用是否显著。

	工 人 甲	工 人 乙	工 人 丙
机器 A	15,15,17	19,19,16	16,18,21
机器 B	17,17,17	15,15,15	19,22,22
机器 C	15,17,16	18,17,16	18,18,18
机器 D	18,20,22	15,16,17	17,17,17

这是双因素试验问题，在命令区运行如下程序：

```
x=[15,19,16;15,19,18;17,16,21;17,15,19;
17,15,22;17,15,22;15,18,18;17,17,18;
16,16,18;18,15,17;20,16,17;22,17,17];
p=anova2(x,3)
```

运行结果如下（其中方差表图如图 31-4 所示）：p =0.0023　0.6645　0.0002。

分析：因为 0.6645>0.05，所以机器因素无显著影响，而 0.0023<0.01，0.0002<0.01，工人因素以及二者的交互因素有非常显著的影响，从数据看，工人丙在机器 B 上产量最高。

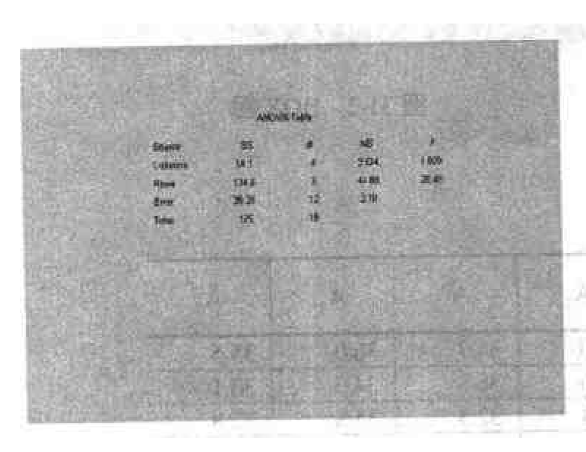


图 31-3　方差表图

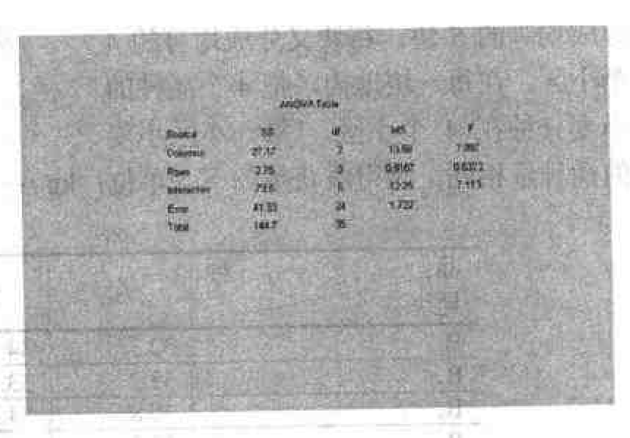


图 31-4　方差表图

【练习小结】

本练习主要介绍了在试验过程中对各个因素对试验结果的影响进行方差分析，并通过方差分析来确定试验过程中各个因素对试验结果影响的程度，从而最终确定影响试验的主要因素

【思考题】

复习本练习中介绍的方差分析的基本思想和在 MATLAB 中的实现方法，并将本练习的例题分别在 MATLAB 中加以实现。

练习 32　信号处理基本函数

知识背景

信号处理工具箱是算法文件的集合，其绝大部分是以 m 文件的形式存在，我们可以用工具箱中的工具包来完成信号处理的功能。信号处理中两个最重要的函数有：filter（对离散数列进行数字滤波），fft（计算数据的离散傅立叶变换）。信号工具箱还利用了其他的标准 MATLAB 函数，如多项式求根、复数运算、矩阵逆以及图形工具等。

主要内容

【本练习考查知识点】

了解信号处理工具箱中的一些基本工具用法，以及用 MATLAB 实现信号处理方面的基本知识。

练习过程

（1）MATLAB 是典型的符号语言，它针对的是矩形数字矩阵，元素可以是实数或者虚数。在信号处理中，一维信号或者序列、多路信号、二维信号等的基本数据对象都可以用矩阵来描述。

当用一维采样数据序列或者信号时，所表示的向量可以为 1×n 或者 n×1 矩阵，其中 n 为采样数。例如在命令区中直接键入：

```
m=[1;3;6;4;8; -2]
```

则有

```
m =
        1
        3
        6
        4
        8
```

−2

用列向量来表示单通道信号，在多通道情况下具有简单、直观的作用。即在多通道中，每一列就表示一个通道，而矩阵中的每一行代表一个采样点。

例如：包含 m，3m，m/3 三个通道的信号可以表示为：

```
x=[m 2*m m/3]
```

则可得到：

```
x =
      1.0000     2.0000     0.3333
      3.0000     6.0000     1.0000
      6.0000    12.0000     2.0000
      4.0000     8.0000     1.3333
      8.0000    16.0000     2.6667
     −2.0000    −4.0000    −0.6667
```

（2）在信号处理中，最重要的有波形的产生。信号处理工具箱中有很多函数可以用于产生不同波形，但其中大部分要求预先输入时间参数。我们在命令框中输入产生采样频率为 500Hz 的时间序列：

```
t=(0:0.002:1)';
```

该命令可以产生 501 个元素的列向量，表示时间从 0~0.5 秒，其中增量为 0.002 秒,符号“'”表示转置，在本例中即将行向量变为列向量。给定 t，假设两个包含两个正弦采样信号的 y：

```
y=sin(2*pi*50*t)+2*sin(2*pi*120*t);
```

再加入白噪声：

```
yn=y+0.5*randn(size(t));
```

画出前 30 个数据的点图，可得如图 32-1 所示。

```
plot(t(1:30),yn(1:30))
```

图 32-1 采样信号图

MATLAB 作为一种程序化语言，可以处理很多的信号，通常我们可以产生不同的信号：

```
y=[1;zeros(49,1)];
y=ones(50,1);
y=2*t;
y=t.^3;
y=square(5*t);
```

包含后两个信号的多通道信号为：

```
z=[t.^3 square(5*t)];
```

而信号工具箱亦能实现多通道单位采样，例：

```
m=[1 1 zero(1,3)]';
```

上式产生的是 5 元素列向量，第一、第二个元素是 1，而剩下的元素是 0。我们也可以

先生成一个包含 1 的行向量，后相乘产生一个矩阵：

```
n=ones(1,3);
x=m*n
```

其结果为：x =

```
1    1    1
1    1    1
0    0    0
0    0    0
0    0    0
```

其结果是一个 5×3 的单位采样矩阵，其中向量 m 中的元素数目是结果矩阵中的行数，而 n 向量中的元素数目是结果矩阵中的列数。

（3）通信中周期波形的产生是一个重要的内容，在 MATLAB 工具箱中有许多函数可以产生不同的周期波形。

① 正余弦波先产生一个采样频率，再采用数学函数计算后作图 32-2 和图 32-3。

```
t=(0:0.01:2*pi)';
x=sin(t);y=cos(t);
plot(t,x),grid;plot(t,y),grid;
```

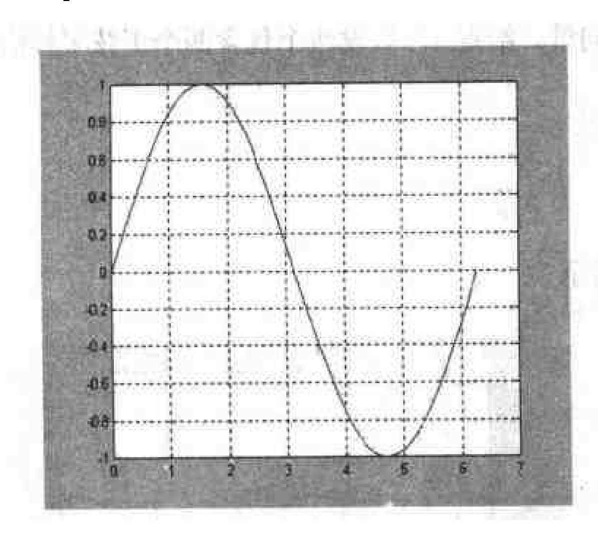


图 32-2　正弦波图

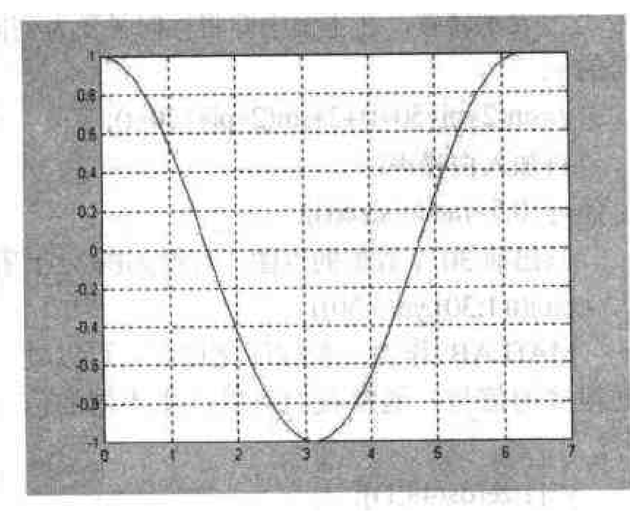


图 32-3　余弦波图

② 三角波

sawtooth(t)　对时间向量 t 产生周期为 2*pi 的锯齿波，与正弦函数波形相似。区别是其波形是锯齿波，峰值为 1(−1)。

sawtooth(t,width)　产生三角波，width 是取值为 0~1 之间的标度参数，决定最大值发生在 0−2*pi 之间的位置，函数值在 0−width*2*pi 的区间由−1 增大到 1，在 width*2*pi−2*pi 区间内从 1 减小到−1。sawtooth(t,1)与 sawtooth(t)相同。

以下语句产生锯齿波，如图 32-4 所示。

```
t=0:0.00001:1;
```

```
x=sawtooth(2*pi*30*t);
plot(t,x),grid
```

③ 方波

square(t)　对时间向量 t 产生周期为 2*pi 的方波，其峰值为 1(−1)。

square(t,duty)　产生特殊方波，其中 duty 是信号为正的区域在一个周期内所占的百分比。如图 32-5 所示。

```
t=0:0.001:1.5;
x=square(2*pi*5*t);
plot(t,x),grid
```

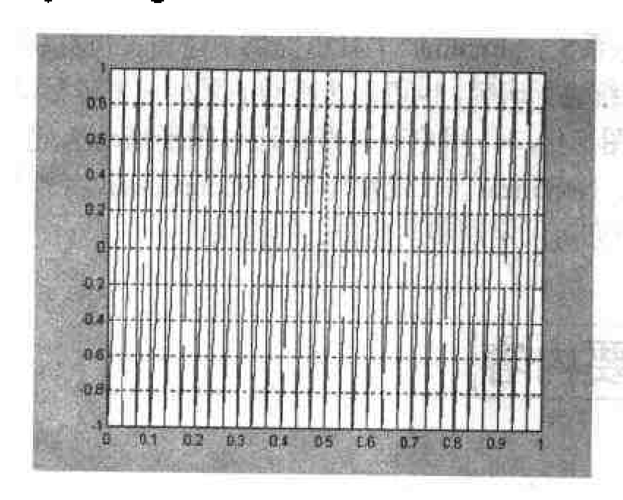


图 32-4　三角波图

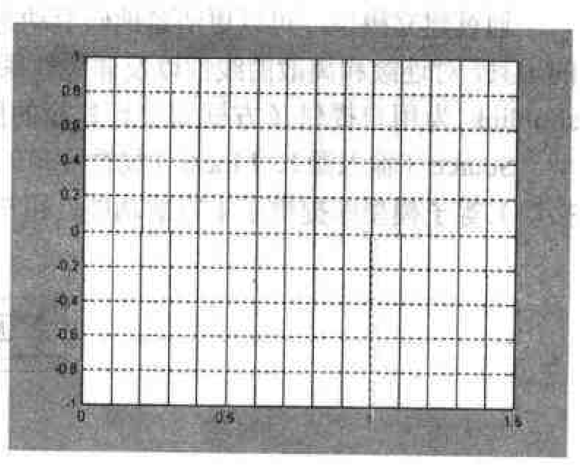


图 32-5　方波图

【练习小结】

本练习介绍在 MATLAB 工具箱信号产生的一些基本函数，以及产生正余弦波、三角波和方波的函数，通过这些函数产生不同频率的波形。在信号处理基本函数中，有两个重要的函数有 filter 和 fft。在信号工具箱中，还有其他的 MATLAB 函数，可以实现不同的功能。

【思考题】

1. MATLAB 中的帮助文件，了解一些基本信号产生的函数以及其产生的波形。
2. 分别根据不同的采样频率（自设）分别在 MATLAB 中生成正余弦波、三角波和方波。

练习 33　动态系统建模仿真

知识背景

通过建立模型，可以更清楚地研究动态系统。simulink 工具箱提供了建模、仿真和分析的工具，对连续和离散的线性以及非线性系统进行研究，也可以对不同的采样速率进行研究。simulink 为用户提供了方框图进行建模的图形接口，用户可以直接通过图形拖动来建立模型。Source（输入源）、Linear（线性环节）、Nonlinear（非线性环节）、Connections（连接与接口）等子模型库接供了丰富的功能，用户亦可以定制和创建自己的模块。

主要内容

【本练习考查知识点】

建立模型后，用户可以通过命令框输入命令来进行仿真，本练习主要通过基本的建模过程来揭示仿真的运用。

练习过程

（1）首先介绍 simulink 工具箱的一些具体操作，我们通过从建模到仿真的过程来揭示 simulink 的使用技巧。

① 在 MATLAB 命令窗口中直接键入 simulink，注意大小写。从而弹出一个窗口，如图 33-1 所示，先新建一个空白窗口，再用鼠标双击模块 simulink 中的子模块 sources，如图 33-2 所示，选择信号发生器 Signal Generator 模块，并将其拖动到空白窗口中。拖动到新窗口中的子模块具有与工具箱一样的缺省设置，如果要定制自己的参数属性，则双击子模块图形，弹出--个属性出口，从而可以输入参数来定制读者所需的子模块。在属性窗口中，可以对 Wave form（波形）、Amplitude（幅值）、Frequency（频率）以及 Units（单位）进行定制。

其中波形可以对下拉菜单进行选择来定制所需的波形，缺省是 sine（正弦波）、square（方波）、sawtooth（三角波）以及 random（随机）。而单位有 Hertz（赫兹）、rad/sec（弧度/秒）。完成参数设置后，可以点“Apply”按钮接交，点键定退出。

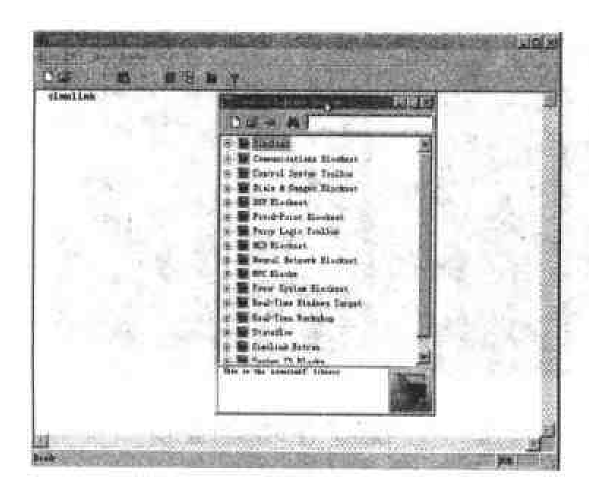


图 33-1 simulink 模块窗口图

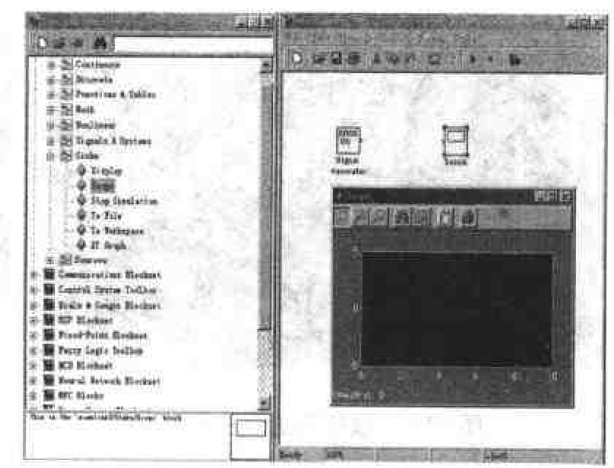


图 33-2 基本模块的拖放

② 选择信号显示模式，从 simulink 工具箱中选择 Sinks 子模块，拖动 Scope 模块到新建窗口。双击 Scope 模块，显示示波器窗口。Signal Generator 和 Scope 图形中的符号>分别表示信号的输入与输出，连接信号时，先将鼠标停在 Signal Generator 的输入端符号上，待鼠标变为十字形后，移动到 Scope 的输入符号后，放开鼠标，即完成连接，结果如图 33-3 所示。此时，一个简单的模型建立完成，我们可以进行一些简单的仿真操作。

（2）仿真过程，我们使用上一个建立的模型，在参数设置完成的情况下，点新建窗口中的 Start 运行，如图 33-4 所示。

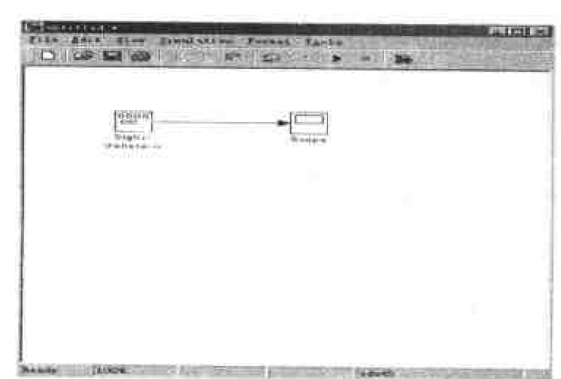


图 33-3 简单模型的建立

图 33-4 模型产生的正弦波

当看不到希望的波形时，需要选择示波器的频率范围和波形幅值范围。当分别选择其他信号源并确定设置参数后，分别运行，如图 33-5 和图 33-6 所示。

（3）算法参数设置，在 simulink 中，仿真中动态数据的计算是由数值积分来实现的。以上的简单模型从数据源直接到示波器，没有通过其他环节，但是动态数据仍然是经数值积分运算而得，因此必须选择适当的参数。我们选择算法参数是从新建窗口的 Simulation 选择项的 Parameters 选项，弹出一个对话框，进而进行参数的设置和选择。

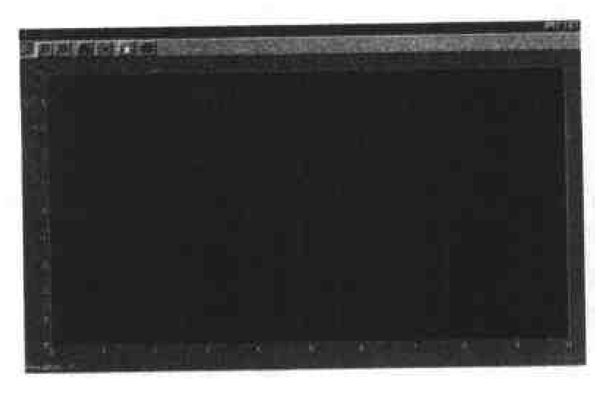


图 33-5 模型产生的三角波

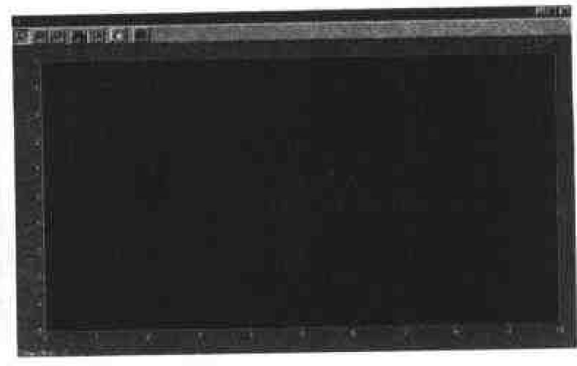


图 33-6 模型产生的随机波形

【练习小结】

本练习主要通过打开 simulink 工具箱，拖放其中的一些模块建立一个简单的模型来研究波形的产生，以及信号输出到模拟示波器上进行显示。由于 simulink 工具箱中提供的模块远不止于此，用户可以根据自己的需要来选择所需的模块建立自己所需要的模型，从而进行模拟仿真。

【思考题】

1. 通过 MATLAB 帮助文件来进一步了解 simulink 工具箱的使用。
2. 试改变参数用 simulink 工具箱来建立模型进行模拟仿真。

练习 34　滤 波 器（一）

知识背景

滤波作为信号处理的重要组成部分，在信号处理中占有很重要的地位，滤波的数学基础是卷积。

主要内容

【本练习讲述知识点】

我们将具体地介绍滤波器，本练习将介绍滤波器的特征，包括脉冲、幅频、群延迟以及零极点位置等，还将介绍工具箱中滤波器函数的用法以及实例。

练习过程

（1）滤波的数学基础是卷积，conv 函数执行一维卷积，在命令窗口中键入：

```
result=conv([2 2 2],[1 1 1])
```

结果为：

```
result =
     2     4     6     4     2
```

对于二维信号，可以使用 conv2 来进行矩阵卷积：

```
a=[1,1,1;2,2,2];
b=[3,2;4,2];
result=conv2(a,b)
```

结果为：

```
resnlt =
     3     5     5     2
    10    16    16     6
     8    12    12     4
```

数字滤波器的输出 y(n)与输入 x(n)之间的关系是单位脉冲响应 h(n)，定义的输出关系：

$$y(n)=\sum_{i=-\infty}^{\infty}h(n-i)x(n)$$

若数字示波器的脉冲响应 h(n)为有限长度，而且输入 x(n)也是有限长度，则可用 conv 来进行滤波，x(n)储存在向量 x 中，h(n)储存在向量 h 中，以下进行卷积计算：

```
x=randn(4,1);
y=[1 3 4 2];
z=conv(x,y)
```

可得运行结果为：

```
z =
   0.3273    1.1565    1.6464    1.5188    1.7798    2.5297    1.4516
```

（2）由于在 MATLAB 中滤波函数 filter 为内建函数，我们通过一个简单的滤波运用来介绍函数。先输入一个简单的单极点低通滤波器：（向量 b 和 a 分别代表传递函数形式滤波器的系数）

```
b=1;
a=[1 -0.5];
y=filter(b,a,x)
```

filter 函数储输出长度与输入长度相同，如果 a 的第一个元素不为 1，则 filter 函数先将各个系数除以 a(1)，运行结果为：

```
y =  0.3273
      0.3383
    -0.0176
      0.7170
```

（3）数字滤波器的脉冲响应是对应于 0 和 1 值的函数序列，在 MATLAB 中有很多方法产生脉冲响应：

```
imp=[1;zeros(19,1)];
```

可得结果为：imp =[1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0]。

对于给定向量的一个简单滤波器为：

```
b=1;
a=[1  -0.9];
h=filter(b,a,imp)
```

运行结果为：h =[1.0000;0.9000;0.8100;0.7290;0.6561;0.5905;0.5314;0.4783;0.4305;0.3874;0.3487;0.3138;0.2824;0.2542;0.2288;0.2059;0.1853;0.1668;0.1501;0.1351]。

使用工具箱可以使上面的过程简化：

```
impz(b,a)
```

从图 34-1 中可以看出，单极点系统按指数 0.9(n)延迟。滤波器中的初始条件和最终条件都很重要，特别是考虑存储限制的时候，输入两个随机序列，序列 a1 对应起始，序列 a2 对应其后，组合序列为 x=[a1,a2]。当没有足够空间存储组合序列来对应 a1 和 a2 来进行顺序滤

波时，为保证序列的连续性，用 a1 最终条件作为滤波器 a2 的初始条件：

```
b=1;
a=[1 －0.9];
a1=randn(1000,1);
a2=randn(1000,1);
[y1,zf]=filter(b,a,a1);
y2=filter(b,a,a2,zf);
```

filtic 函数为 filter 产生初始条件，filtic 安排延迟向量使得滤波器的行为反映过去的输入和输出的影响，例：

```
zf=filtic(b,a,flipud(y1),flipud(a1));
```

此时获得的延迟向量 zf 与上相同，这一方式对于较短的数据序列进行滤波时有用。

（4）除了 filter 函数外，其他滤波函数还有 filtfilt 和 fftfilt。filtfilt 在滤波过程中消除相位失真，对于 IIR 滤波，相位失真通常为强非线性，filtfilt 函数使用信号中当前点前后的数据点信息，本质上为消除相位失真。

例：采样频率为 100Hz 的一个信号，它由频率为 3Hz 和 40Hz 的正弦信号组成。创建一个 10 点的 FIR 滤他器，使用 filter 和 filtfilt 函数进行比较：

```
t=0:0.01:1;
x=sin(2*pi*t*3)+sin(2*pi*t*40);
b=ones(1,10)/10;
y=filtfilt(b,1,x);
z=filter(b,1,x);
plot(t,x,t,y,'－',t,z,':')
```

三个滤波器的运行结条比较如图 34-2 所示。

滤波效果消除了原来信号中频率为 40Hz 的分量。虚线（filtfilt）与原信号中 3Hz 一致，而 filter 滤波结果要延迟大约 5 个采样点。

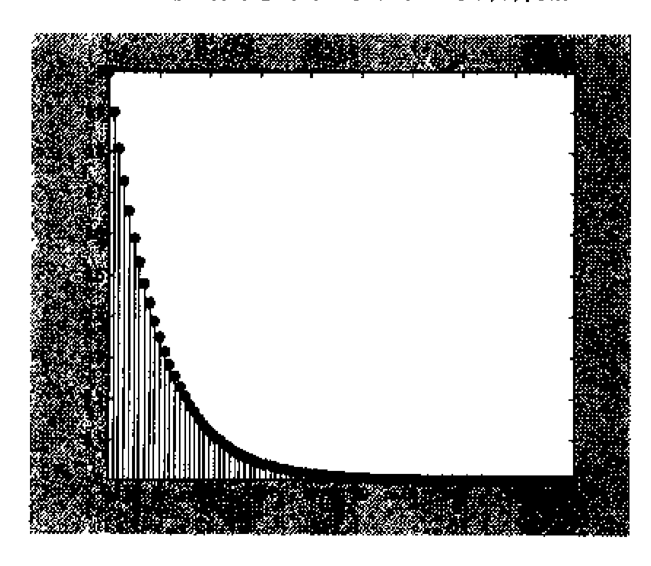


图 34-1　滤波信号图

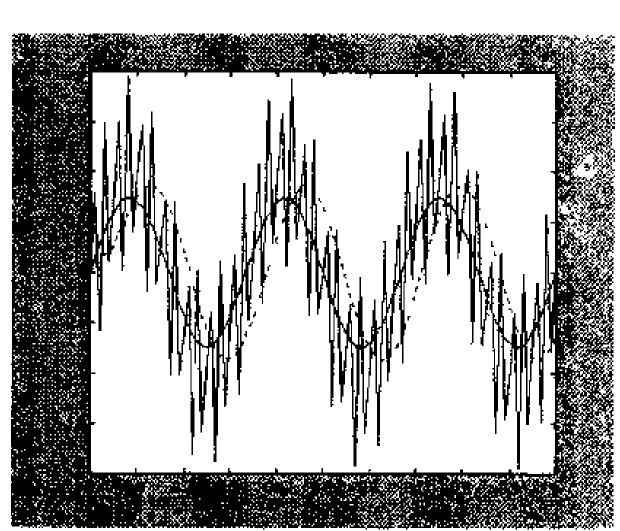


图 34-2　三种滤波器运行比较

【练习小结】

本练习介绍了滤波器的特点，而滤波的数学基础是卷积，练习先介绍了卷积在 MATLAB 中的实现函数 conv。接着介绍了脉冲、幅频、群延迟的概念和思想，最后介绍了三种滤波器分别的实现函数和它们的使用与区别。

【思考题】

1. 复习本练习中介绍的滤波的基本思想和函数实现。
2. 熟练三种滤脉器的实现函数和它们的使用，并在 MATLAB 命令窗口中加以实现。

练习 35 滤 波 器（二）

知识背景

滤波作为信号处理的重要组成部分，在信号处理中占有很重要的地位，滤波的数学基础是卷积。

主要内容

【本练习考查知识点】

信号处理工具箱中有很多工具函数对模拟和数字滤波器进行频率域分析，其中函数 freqz 和 freqs 分别返回数字和模拟滤波器的复频率响应。

练习过程

（1）函数 freqz 使用基于 FFT 的算法计算数字滤波器的频率响应，使用命令方法：

```
[p,q]=fredz(b,a,n)
```

fredz 包含滤波器系数向量 b 和 a，整数 n 是频率响应的点数，fredz 返回复频率响应 p 和频率点 q。fredz 能包含一些参数，可以包含采样频率和任意的采样频率，以下对 12 阶切比雪夫 I 型滤波器求其 256 点频率响应，采样频率 F 为 1000Hz：

```
[m,n]=cheby1(12,0.5,200/500);
[h,f]=freqz(m,n,256,1000);
```

运行所得结果为：（略）

由于参数表包括采样频率，故 freqz 在计算频率响应时，需存在一个包含在 0 到 F/2 之内有 256 个频率点的向量 f。若调用 fredz 函数而不加输出说明，则自动画出滤波器的副频响应。设一个截止频率为 400Hz，采样频率为 2000Hz 的 9Butterworth 低通滤波器，对此滤波器设计其 256 点复频率响应，画出其幅值图与相位图，如图 35-1 所示。

```
[p,q]=butter(9,400/1000);
freqz(p,q,256,2000)
```

所得结果如图 35-1 所示。

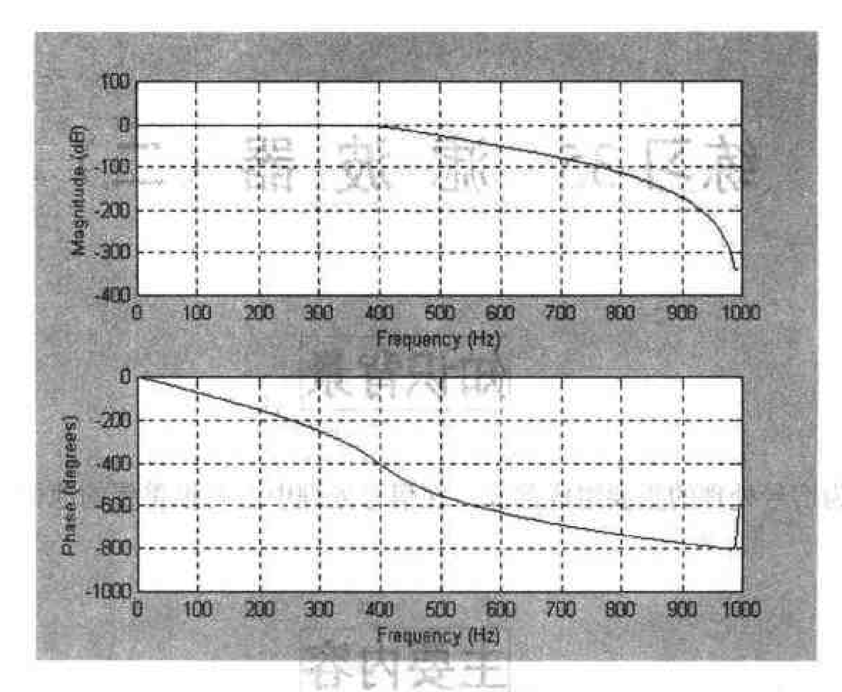


图 35-1　幅值图与相位图

freqz 还能处理任意频率点的频率响应，如：

```
w=linspace(0,pi);
h=freqz(p,q,w)
```

所得结果为：(略)

计算由向量 p、q 定义的滤波器在由 w 指定的频率点的复频率响应。频率点的范围可以从 0－2*pi，为了指定范围从 0 到采样频率的频率向量，在参数表中要包括频率向量和采样频率值。

（2）函数 freqs 能为由两个输入系数向量 p 和 q 定义的模拟滤波器计算频率响应，其用法与 freqz 类似，可以指定频率点的数目(缺省值为 200)，提供任意的频率点向量，并画出幅值和相位响应。

（3）在 MATLAB 中提供了从频率响应向量 h 中提取幅值和相位的函数，函数 abs 返回响应的幅值，函数 angle 返回响应的相位，单位为 rad。例从 Butterworth 滤波器中提取并画出幅值与相位图：

```
[p,q]=butter(6,300/500);[h,w]=freqz(p,q,512,1000);
m=abs(h);r=angle(h);
semilogy(w,m);
plot(w,m);
plot(w,r*180/pi)
```

得到的结果如图 35-2 和图 35-3 所示。

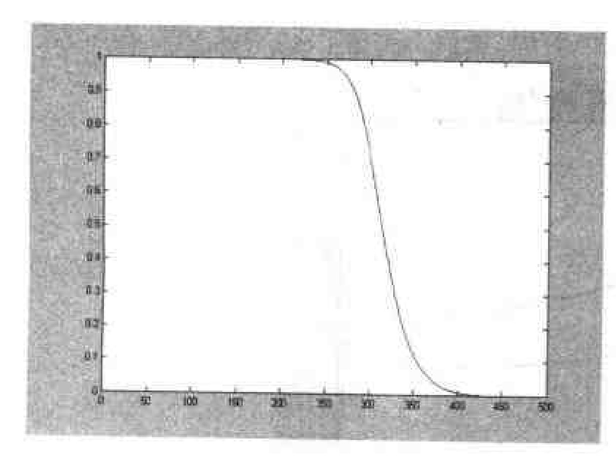


图 35-2　Butterworth 滤波器幅值图

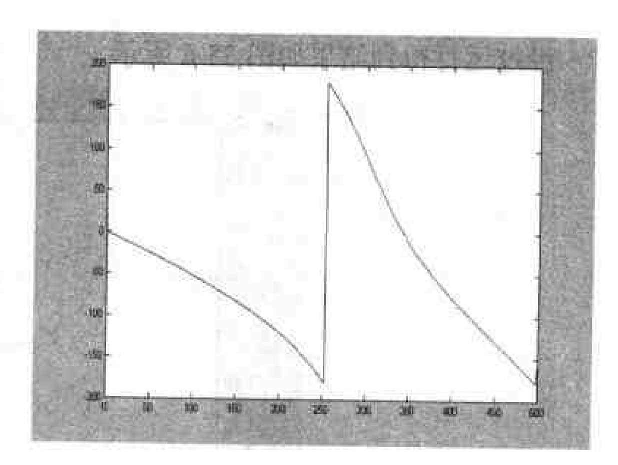


图 35-3　Butterworth 滤波器相位图

在通信的频率分析中，函数 unwrap 是一个很有用的函数，其通过加上 360 度的倍数来展开相角使之穿越 180 度边界线连续。例设计一个 25 阶低通滤波器，用 freqz 得到其频率响应，画出其相位图(相位单位为度)，采用 unwrap 消除 360 度跳跃：

```
h=fir1(25,0.4);
[H,f]=freqz(h,1,512,2);
plot(f,angle(H)*180/pi);grid
plot(f,unwrap(angle(H))*180/pi);
grid
```

所得结果如图 35-4 和 35-5 所示。

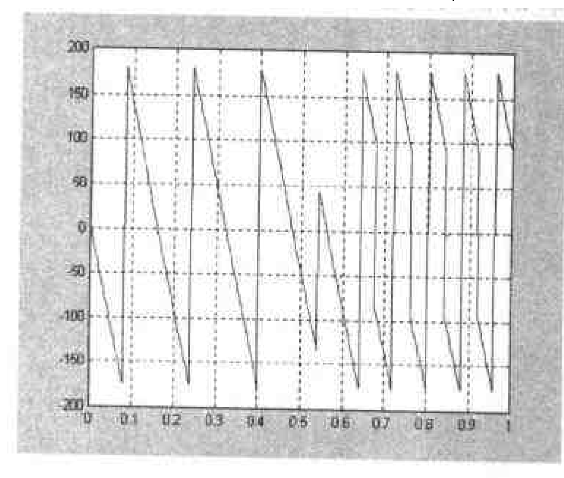


图 35-4　低通滤波器相位图

图 35-5　经函数 unwrap 处理后的相位图

（4）在 MATLAB 中计算群延迟：

```
[u,v]=grpdelay(b,a,n)
```

函数所返回由 b 和 a 定义的滤波器的 n 点群延迟。在图中画出系统的群延迟和相位延迟：

```
[b,a]=butter(10,200/1000);
u=grpdelay(b,a,128);
[h,f]=freqz(b,a,128,2000);
pd=－unwrap(angle(h))*(2000/(2*pi))./f;
plot(f,u,'－',f,pd,'r')
```

经过运行得到结果如图 35-6 所示。

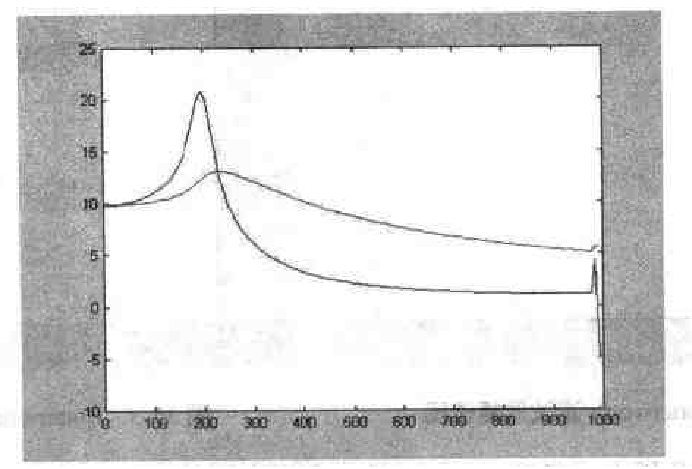


图 35-6 群延迟和相位延迟图

【练习小结】

本练习主要对模拟和数字滤波器进行频率域分析，通过运用实例来说明函数 freqz 和 freqs 返回数字和模拟滤波器的复频率响应。

【思考题】

复习本练习介绍的两个主要函数 freqz 和 freqs 的相关概念和用法。

练习 36 小 波 分 析

知识背景

小波分析是强有力的信号分析工具，小波分析作为泛函分析、傅立叶分析、样条理论、调和分析和数值分析等多个学科交叉而成。小波分析是多尺度信号分析方法，是分析非平稳信号的有力工具，它克服了傅立叶变换固定分辨率的特点，既可以分析信号的概貌，又可分析信号的细节。

主要内容

【本练习考查知识点】

MATLAB 工具箱中集中了很多小波分析的工具，有很多分析小波的工具包和函数，还有可视化的分析界面，为研究人员提供了设计和仿真以及运用的平台。本练习主要介绍小波分析的基本概念以及在工具箱中的运用和分析的主要工具。

练习过程

（1）在小波分析中，需要用到一些常用的基函数，而 MATLAB 工具箱中提供了实貌基函数的工具包。在使用这些函数时，可以通过设置参数来选样用户所需要的小波基函数。

主要的小波函数为：

morl－Morlet 小波；

mexh－墨西哥草概小波；

meyr－meyer 小波；

haar－Haar 小波；

dbN－紧支撑双正交小波；

coifN－Coifmant 小波；

biorNr.Nd－双正交样条小波。

（2）MATLAB 分析工具概提供了计算小波滤波器系数的函数，下面分别举例计算。

① 计算双正交样条小波尺度函数滤波器系数的主要函数为 biorwavf(x)，其用法为：

[a,b]=biorwavf(x)

其中[a,b]分别返回双正交样条小波的分解滤波器和重构滤波器，x 为双正交小波系数。例：

在命令区中键入：

```
wname='bior2.2';
[a,b]=biorwavf(wname)
```

运行结果为：

```
a =0.2500    0.5000    0.2500
b =－0.1250   0.2500    0.7500    0.2500   －0.1250
```

② 计算 Coifwavf 小波尺度滤波器的系数所用的函数为 Coifwavf，其用法为：

h=coifwavf(x)

其中 h 返回 Coifwavf 小波的尺度滤波器，而 x='coifN'为 Coifwavf 小波的系数，N 值可以选 1、2、3、4、5。例：

```
x='coif2';h=coifwavf(x)
```

运行结果为：

```
h =Columns 1 through 7
 0.0116   －0.0293   －0.0476    0.2730    0.5747    0.2949   －0.0541
Columns 8 through 12
－0.0420    0.0167    0.0040   －0.0013   －0.0005
```

③ 计算紧支撑双正交小波尺度滤波器系数，主要函数为 dbaux，用法为：

s=dbaux(N,SUMW)

其中 s 返回的是 N 阶 Daubechies 小波尺度滤波器，满足 sum(s)=SUMW，SUMW 为可选项，缺省为 1。例：

```
eg=[－0.0625 0 0.5625 1 0.5625 0 －0.0625];
rg=roots(eg);
cw=poly([rg(6) －1 －1]);
cw=cw/sum(cw);s=dbaux(2)
```

运行后得到结果为：

```
s =0.3415    0.5915    0.1585   －0.0915
```

④ 计算紧支撑双正交小波滤波器系数，函数为 dbwavf(x)，用法为：

p=dbwavf(x)

其中 p 返回为字符串 x 所确定的紧支撑双正交小波滤波器，x='dbN'，N 的取值为 1、2……。例：

```
x='db3';
p=dbwavf(x)
```

⑤ 计算墨西哥小帽小波滤波器系数的函数是 mexihat，用法为：

[PSI,x]=mexihat(l,u,N)

其中 PSI 返回区间[l,u]中长度为 N 的格点 X 上的墨西哥小帽小波滤波器。

例：计算区间[－3,3]上的墨西哥小帽小波滤波器，长度为 500。

在命令区中输入下列命令：

```
l=－3;u=3;
n=500;
[psi,x]=mexihat(l,u,n);
plot(x,psi)
```

运行后得到结果如图 36-1 所示。

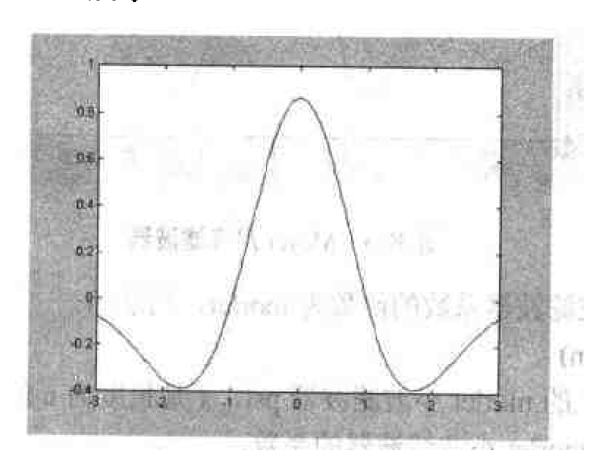


图 36-1 墨西哥小帽滤波器图

⑥ 计算 Meyer 小波滤波器及其尺度滤波器系数的函数为 meyer，用法为：

```
[phi,psi,t]=meyer(lb,ub,n)
```

它返回区间[lb,ub]中长度为 n 的格点 t 上的 Meyer 小波滤波器 phi 和尺度滤波器 psi。

例：计算[－4，4]上长度为 512 的 Meyer 小波滤波器及其尺度滤波器。

```
lb=－4;ub=4;n=512;
[phi,psi,x]=meyer(lb,ub,n);
plot(x,psi);
plot(x,phi)
```

运行后得到结果如图 36-2 和图 36-3 所示。

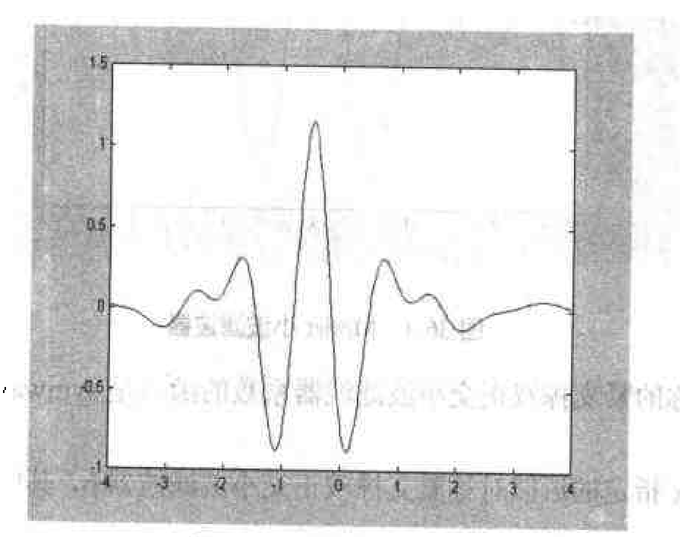


图 36-2 Meyer 小波滤波器

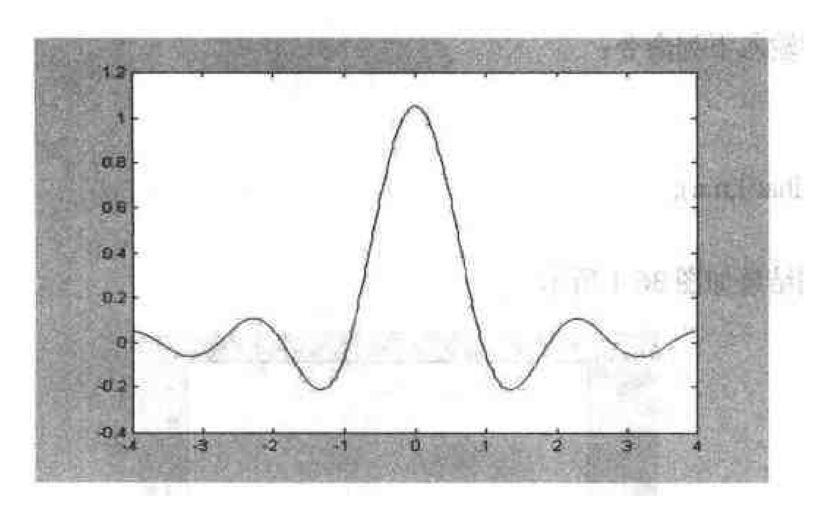


图 36-3 Meyer 尺度滤波器

⑦ 计算 morlet 小波滤波器系数的函数为 morlet，用法为：

[psi,x]=morlet(lb,ub,n)

它返回区间[lb,ub]上的 morlet 小波滤波器 psi，x 是长度为 n 的格点。

例计算[－3,3]上的 morlet 小波滤波器的系数。

```
lb=－3;ub=3;
n=500;
[psi,x]=morlet(lb,ub,n);
plot(x,psi)
```

运行后得到结果如图 36-4 所示。

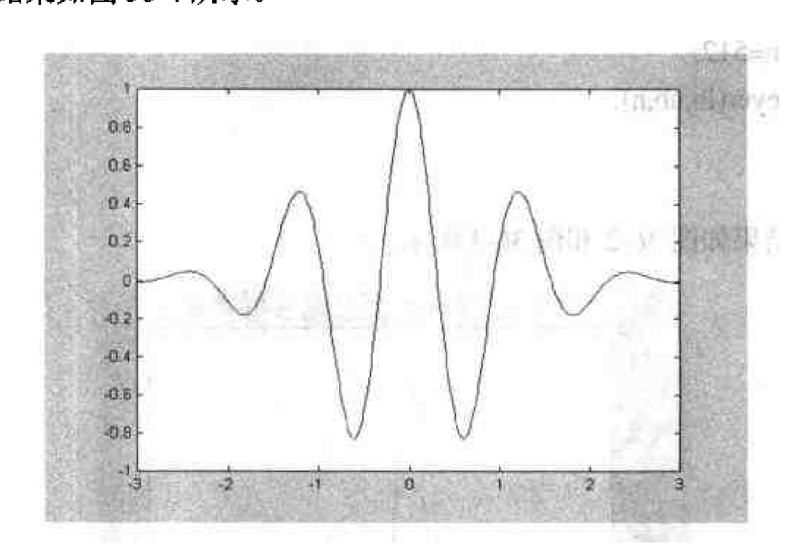


图 36-4 Morlet 小波滤波器

⑧ 计算近似对称的紧支撑双正交小波滤波器系数的函数是 symwavf，用法为：

b=symwavf(x)

它返回字符串 x 指定的近似对称紧支撑双正交小波滤波器 b，其中 x='symN'，其中 N=1、2、3……。例：

```
b=symwavf('sym4')
```

运行后得到：

```
b =Columns 1 through 7
   0.0228   -0.0089   -0.0702   0.2106   0.5683   0.3519   -0.0210
  Column 8
  -0.0536
```

【练习小结】

本练习主要介绍了小波分析中的各个基函数和计算各种小波滤波器系数的函数，并在一定条件下给出波形来研究系数。

【思考题】

1．回顾本练习中讲述的各种小波基函数和它们的基本概念。

2．在 MATLAB 中分别将各个函数加以实现。

练习 37　小波去噪与压缩

知识背景

小波分析中最重要的两个运用是信号去噪和信号压缩，使用小波分解可以将原信号分解为一系列的近似分量和细节分量，信号的噪声集中表现在细节分量。使用一定的与阈值处理细节分量，再经过小波重构后可以得到平滑的信号。信号的小波压缩与小波去噪在本质上是相同的，即经过小波重构后可以得到平滑的信号。小波分析与小波去噪的主要区别在于阈值的选择算法不同，可以使用全局阈值，也可以使用自适应阈值。

主要内容

【本练习讲述知识点】

本练习主要讲述小波分析中信号去噪和信号压缩的基本思想，以及进行相应处理的相关函数，主要通过函数的用法实例和 MATLAB 自带的实例来揭示小波分析中信号的处理方法。由于小波分析中所用到的函数众多，本练习只是给出常用和比较重要的分析函数和工具包。对于具体的需要，读者可以通过 MATLAB 的帮助文件来获得详细资料。

练习过程

（1）小波去噪和压缩阈值选取方案可以通过函数 ddencmp，其用法为：

[THR,SORH,KEEPAPP,CRIT] = ddencmp(dorc,worwp,x)

可以有三种使用形式：

① [THR,SORH,KEEPAPP,CRIT] = DDENCMP(IN1,IN2,X)

② [THR,SORH,KEEPAPP] = DDENCMP(IN1,'wv',X)

③ [THR,SORH,KEEPAPP,CRIT] = DDENCMP(IN1,'wp',X)

根据信号 x 和参数 IN1 和 IN2 自动生成小波去噪和压缩的阈值选取方案。其中 IN1 可选值为“den”为信号去噪，“cmp”为信号压缩。IN2 可选值为“wv”使用小波分解，取值为“cmp”时使用小波包分解。而 THR 为函数选择的阈值，SORH 为 ddencmp 函数选择的阈值使用方式，取值为“s”时为软阈值，取值为“h”时为硬阈值。而 KEEPAPP 决定了是

否对近似分量进行阈值处理，可选 1 或者 0。CRIT 为使用小波包分解时选取的熵函数类型。例：

```
init=21564148656;randn('seed',init);
x=randn(1,800);
[thr,sorh,keepapp]=ddencmp('den','wv',x)
[thr,sorh,keepapp]=ddencmp('cmp','wv',x)
[thr,sorh,keepapp]=ddencmp('den','wp',x)
[thr,sorh,keepapp]=ddencmp('cmp','wp',x)
```

在不同的信号处理方式下所得结果如下：

```
thr =
    3.68515752973086
sorh =
    s
keepapp =
        1
thr =
    0.67980588564722
sorh =
    h
keepapp =
        1
thr =
    4.23105962519793
sorh =
    h
keepapp =
        1
thr =
    0.67980588564722
sorh =
    h
keepapp =
        1
```

（2）当选取用于小波去噪的阈值时，可以使用函数 thselect，其用法为：

```
thr = thselect(x,tptr)
```

其中 thr 为返回用于小波去噪的阈值，x 为有噪信号。参数 tptr 则指定了阈值选取算法，可选值分别为：rigrsure、heursure、sqtwolog、minimaxi。举例如下：

```
init=21564148656;randn('seed',init);
x=randn(1,800);
thr=thselect(x, 'rigrsure')
```

```
thr=thselect(x, 'heursure')
thr=thselect(x, 'sqtwolog')
thr=thselect(x, 'minimaxi')
```

在参数取值不同的情况下所得结果分别为：

```
thr =
     1.99908916578119
thr =
     3.65639487136385
thr =
     3.65639487136385
thr =
     2.15746129710980
```

（3）一维信号的小波去噪处理的函数主要为 wden，函数形式为：

```
[xd,cxd,lxd] = wden(in1,in2,in3,in4,in5,in6,in7)
```

主要运用方式为：

```
[XD,CXD,LXD] = WDEN(X,TPTR,SORH,SCAL,N,'wname')
[XD,CXD,LXD] = WDEN(C,L,TPTR,SORH,SCAL,N,'wname')
```

函数处理后返回去噪处理的信号 xd，及其小波分解结构[cxd,lxd]，X 为有噪信号。参数 tptr 指定阈值选取算法，其可选值为：rigrsure、heursure、sqtwolog、minimaxi。sorh 为函数选择的阈值使用方式，当取值为's'时，为软阈值；取值为'h'时,为硬阈值。scal 可以取的参数值为：one、sln、mln；wname 为小波基函数的名称。而[xd,cxd,lxd] = wden(c,l,tptr,sorh,seal,n,'wname')由有噪声信号的小波分解结构[c,l]得到去噪后的信号 xd。我们用 MATLAB 自带的实例来举例说明：

```
snr=3;init=2055615866;
[xref,x]=wnoise(3,11,snr,init);
lev=5;
wd=wden(x,'heursure','s','one',lev,'sym8');
subplot(321),plot(xref),axis([1,2048, －10,10]);
title('Original signal');
subplot(322),plot(x), axis([1,2048,－10,10]);
title(['Noisy signal-Signal to noise ratio=',num2str(fix(snr))]);
subplot(323),plot(xd),axis([1,2048,－10,10]);
title('De-noised signal-heuristic SURE');
xd=wden(x,'heursure','s','one',lev,'sym8');
   subplot(324),plot(xd),axis([1,2048,－10,10]);
   title('De-noised signal-SURE');
   xd=wden(x,'sqtwolog','s','sln',lev,'sym8');
   snbplot(325),plot(xd),axis([1,2048,－10,10]);
   title('De-noised signal-Fixed form threshold');
   xd=wden(x,'minimaxi','s','sln',lev,'sym8');
```

```
subplot(326),plot(xd),axis([1,2048,－10,10]);
title('De-noised signal-minimax');
[c,l]=wavedec(x,lev,'sym8');
xd=wden(c,l,'minimaxi','s','sln',lev,'sym8');
```

运行结果如图 37-1 所示。

（4）信号的小波去噪和小波压缩的函数是 wdencmp，其函数形式为：

[xc,cxc,lxc,perf0,perfl2] = wdencmp(o,varargin)

主要运用有：

[XC,CXC,LXC,PERF0,PERFL2] =WDENCMP(Y,X,'wname',N,THR,SORH,KEEPAPP)

wdencmp函数用于一维或者二维信号的压缩和去噪，通过不同的阈值选择来选取方案。函数返回处理后的信号xd及其小波分解结构[cxd,lxd]，perf0和perfl2为采用2范数度量的信号恢复率和压缩率。Y可以选择“gbl”或者“lvd”，N为分解的层数。keepapp取值为1时，保持近似分量不变；当keepapp取值为0时，对近似分量进行调值处理。我们仍以MATLAB自带的实例来练习：

图像去噪：

```
load sinsin
init=2055615866;randn('seed',init);
  x=X+18*randn(size(X));
  [thr,sorh,keepapp]=ddencmp('den','wv',x);
  xd=wdencmp('gbl',x,'sym4',2,thr,sorh,keepapp);
  subplnt(1,3,1),imshow(X,[])
  subplot(1,3,2),imshow(x,[])
  subplot(1,3,3),imshow(xd,[])
```

所得结果见图 37-2（三个图分别为原图像、有噪图像和去噪后的图像）。

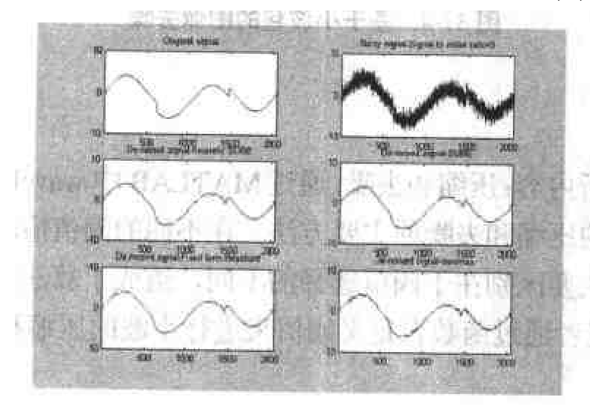


图 37-1　采用不同阈值的小波去噪

图 37-2　图像的小波去噪

图像压缩：

```
load woman
x=X(100:200,100:200);nbc=size(map,1);n=5;w='sym2';
[c,l]=wavedec2(x,n,w);thr_h=[17 18];thr_d=[19 20];thr_v=[21 22];
thr=[thr_h;thr_d;thr_v];[xd,cxd,lxd,perf0,perfl2]=wdencmp('lvd',x,'sym8',2,thr,'h');
```

```
subplot(1,2,1),imshow(x,[])
subplot(1,2,2), imshow(xd,[])
```

所得图像为如图 37-3（两个图像分别为原图像和压缩、解压缩后的图像）。

（5）基于小波包的信号去噪或压缩的主要函数为 wpdencmp，其用法主要为：

```
[XD,TREED,DATAD,PERF0,PERFL2] =
    WPDENCMP(X,SORH,N,'wname',CRIT,PAR,KEEPAPP)
[XD,TREED,DATAD,PERF0,PERFL2] =
    WPDENCMP(TREE,DATA,SORH,CRIT,PAR,KEEPAPP)
```

参数和函数返回值与前介绍的基本一致，我们用 MATLAB 自带的实例来说明：

```
load sinsin
init=2055615866;randn('seed',init);
x=X+18*randn(size(X));
[thr,sorh,keepapp,crit]=ddencmp('den','wp',x);
xd=wpdencmp(x,sorh,3,'sym4',crit,thr,keepapp);
subplot(1,3,1),imshow(X,[]),subplot(1,3,2),imshow(x,[]),subplot(1,3,3),imshow(xd,[])
```

运行得到图像如图 37-4 所示（三个图分别为原图像、有噪图像和去噪后的图像）。

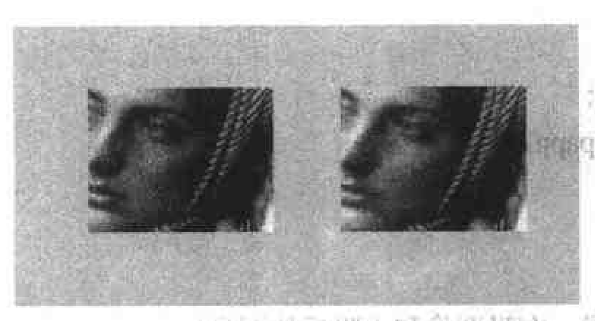


图 37-3 小波图像压缩

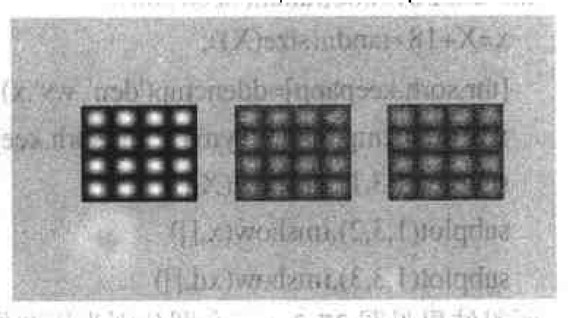


图 37-4 基于小波包的图像去噪

【练习小结】

本练习介绍了在小波分析中重要的两个分析内容：压缩和去噪，通过 MATLAB 中 wavelet 工具箱相关的压缩和去噪函数来说明进行小波的压缩和去噪的主要方法。在不同的阈值情况下，小波进行不同的处理。而小波压缩与去噪主要区别在于阈值选择的不同，造成了算法的不同。在 MATLAB 中，用不同的函数自适应或者通过函数自定义阈值来进行小波的压缩和去噪处理。

【思考题】

请将练习中介绍的实例在 MATLAB 实现，并可以通过其自带的一些帮助文件以及实例来近一步了解小波分析去噪和压缩的功能。

练习 38　信号变换

知识背景

在信号处理工具箱中，有比较重要的几类变换函数：离散傅立叶变换 DFT、Chirp-z 变换、离数余弦变换 DCT 和 Hilbert 变换等。离散傅立叶变换是单位圆上的 z 变换，此变换使离散序列的描述和处理相对变得容易。Chirp-z 变换在沿轮廓线而非单位圆的 z 变换中很有用，在计算 prime-length 变数时，Chirp-z 变换比 DFT 算法效率更高。离散余弦变换 DCT 与 DFT 比较接近，其能量压缩特性在信号代码运用中非常有用。Hilbert 变换能简化解析信号的格式，而解析信号广泛使用在通讯领域的带通信号处理中。

主要内容

【本练习讲述知识点】

本练习将介绍信号变换的基本思想和信号变换在 MATLAB 中的具体运用，并通过实例介绍信号变数在 MATLAB 实现的函数和工具包。

练习过程

（1）离数傅立叶变换(DFT)实现的主要函数是 dftmtx，而快速离散傅立叶变换(FFT)是信号处理工具箱的基础，是对 DFT 进行快速计算的一种有数算法。计算离数傅立叶变换矩阵的函数 dftmtx 的用法为：

A=dftmtx(x)

其中函数返回值为 x*x 的 DFT 变换矩阵，而离散傅立叶反变数(IDFT)矩阵为 Ai=conj(dftmtx(x))/x

工具箱提供了 fft 函数和 ifft 函数分别计算快速离数傅立叶变换及其逆变换。对于单输入 x，函数 fft 用来计算输入向量或者矩阵的 DFT。若 x 为向量 fft 计算向量的 DFT，若 x 为矩阵，fft 计算每一列的 DFT。函数 fft 的用法为：

y=fft(x)

y=fft(x,n)

y=fft(x)计算信号 x 的快速离散傅立叶变换，y=fft(x,n)计算 n 点 FFT。若按 y=fft(x,n)的方法计算时，n 代表 DFT 的 长度，当输入序列比 n 短时，函数 fft 用 0 填充输入序列，反之则截短傳入序列。若 n 值缺省，则默认 fft 长度为输入序列的长度。fft 的执行时间决定于 DFT 的长度。对于任意为 2 的幂数的 n，fft 使用高速 radix-2 算法，此时执行时间短。对于任意非 2 的幂数 n，fft 使用主因子法，此算法的速度取决于点数 n 的大小和它具有的主因子的数目。对于任意质数 n，fft 不能使用 FFT 算法，它直接执行较慢的方法计算 DFT。逆离散傅立叶逆变换函数 ifft 也包含输入序列、输入选项、进行变换的期望点数等参数。二维傅立叶变换及其逆变换的函数是 fft2 和 ifft2，这些函数用于二维的信号处理或者图傳处理。有时需将 fft 或者 fft2 的输出进行重排以使其零频部分处于序列的中心，使用函数 fftshift 可以将零频部分移到向量或者矩阵的中心。下面举例说明。

例：sinc 函数与方波之间的傅立叶变换关系。

在命令区中傳入：

```
t=0:0.01:1;
x=sinc(2*pi*5*t);
a=dftmtx(length(t));
y=x*a;
plot(t,x)
plot(t,y)
```

运行所得输出图形如图 38-1 所示。

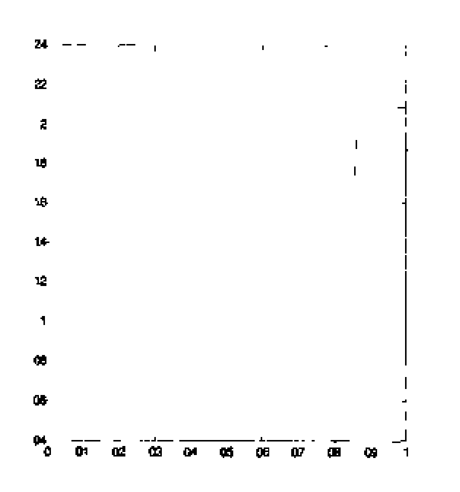


图 38-1　Sinc 函数的 DFT 变换

图 38-2　FFT 信号变换的幅值和相位

例：计算输入信号的 FFT 幅值和相位。

在命令窗口中傳入：

```
t=0:1/99:1;

x=sin(2*pi*15*t)+sin(2*pi*40*t);
y=fft(x);m=abs(y);
p=unwrap(angle(y));
```

```
f=(0:length(y)-1)*99/length(y);
subplot(1,2,1),plot(f,m)
subplot(1,2,2),plot(f,p*180/pi)
```

运行后得到信号的 FFT 幅值和相位，如图 38-2 所示。

例：计算方波的傅立叶反变换。

在命令区中键入：

```
x=[1 1 1 1 0 0 0 0];
y=fftshift(ifft(x,256));
subplot(1,2,1),plot(x),axis([0 8 0 1.5])
subplot(1,2,2),plot(abs(y))
```

所得结果见图 38-3 所示。

从下面这个例子可以发现原始序列与重建序列是等同的：

```
t=(0:1/255:1);
x=sin(2*pi*120*t);
y=real(ifft(fft(x)));
subplot(1,2,1),plot(t,x)
subplot(1,2,2),plot(t,y)
```

由图 38-4 可见两个图形是一样的：

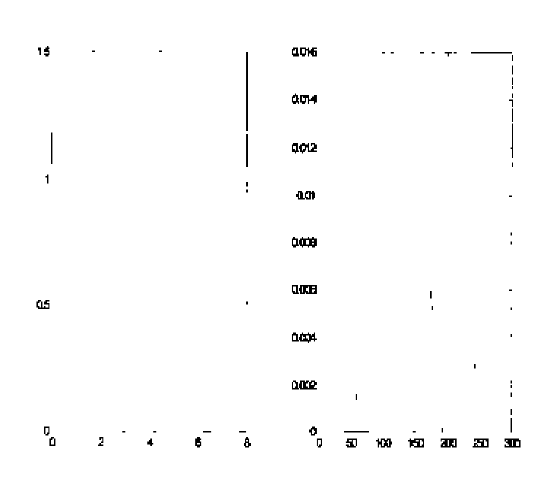


图 38-3 方波的傅立叶反变换

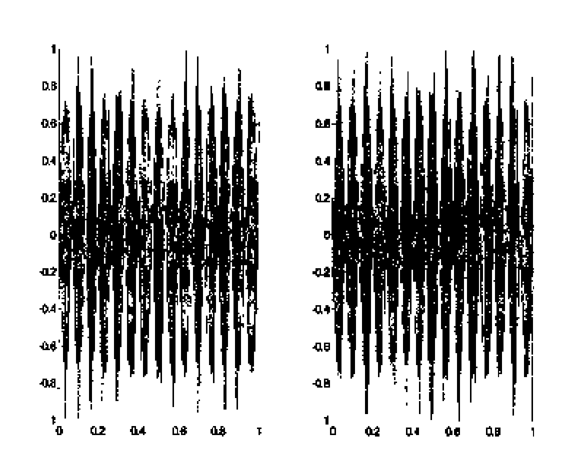


图 38-4 原始序列与重建序列比较图

（2）Chrip-z 变换(CZT)，是指在 z 域内对输入序列沿螺旋轮廓线进行 z 变换。而离散傅立叶变换(DFT)，可以看作是信号在 z 域上沿单位圆的均匀采样，但是在实际运用中，并非整个单位圆上的频谱都有意义。与 DFT 这种计算整个频谱的算法不同，Chirp-z 变换(CZT)是一种更灵活的计算频谱的算法，它并不局限在单位圆上，而是可沿如下描述的轮廓线进行 z 变换：

$$z_t = AW^{-t} \qquad t=0,1,\cdots\cdots,M-1$$

其中 A 是复数起点，W 是描述轮廓线上点间复比率的复数标量，M 是变换的长度，一个可能的螺旋线是

```
a=0.6*exp(j*pi/6);
w=0.9*exp(-j*pi*0.05);
m=80;
z=a*(w.^(-(0:m-1)));
zplane([],z)
```

画出的螺旋线图形如图 38-5 所示。

计算 Chrip-z 变换，MATLAB 工具箱中提供了函数 chirp，其用法为：Y = CHIRP(T,F0,T1,F1)、Y = CHIRP(T,F0,T1,F1,'method') 和 Y = CHIRP(T,F0,T1,F1,'method', PHI)。而更简便的函数是 czt，其用法为：y=czt(x,m,w,a)和 y=czt(x)。其中前一个形式计算指定参数 m，w，a 的 chrip-z 变换，后一个是使用缺省值的 chrip-z 变换，各参数的缺省值为：m= length(X)，w = exp(• j*2*pi/M)和 a=1。

（3）dct 函数（DCT）计算输入向量或者矩阵的离散因果变换。DCT 与离输傅立叶变换很接近，DFT 实际上是计算序列 DCT 的一步，DCT 具有更好的能量压缩性，仅用几个变换系数即可代表序列能量的总体。DCT 的输量压缩性使得它在数据通讯中非常有用。而 idct 函数对输入序列计算逆 DCT 变换，从完全或者部分的 DCT 系数来重建信号是可能的。下面举例说明：

例：先产生一个频率为 10Hz 的正弦信号，其采样频率为 1000Hz：

```
t=0:1/999:1;
x=sin(2*pi*25*t);
```

后计算此序列的 DCT 并仅用值大于 53 的部分重建信号，画出原始和经过重建的信号：

```
y=dct(x);
y2=find(abs(y)<53);
y(y2)=zeros(size(y2));
z=idct(y);
plot(t,x);
plot(t,z)
```

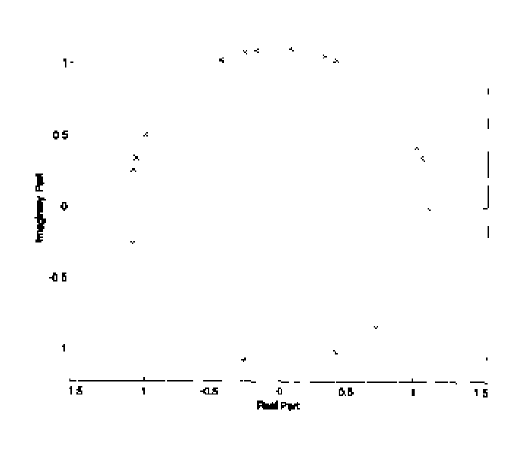


图 38-5 螺旋线图材

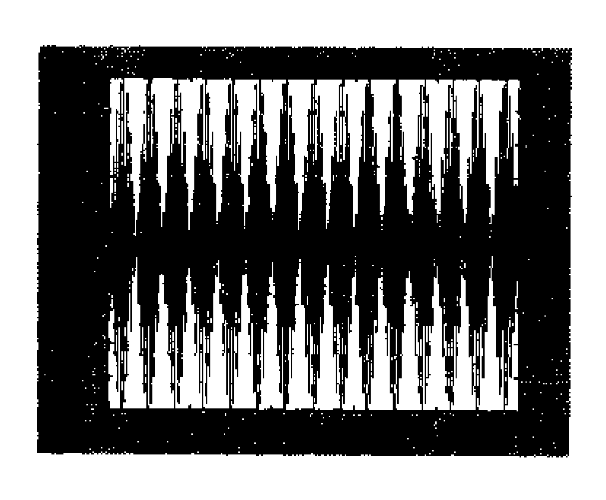


图 38-6 采样原始信号图

原始信号和经过重建的信号图形如图 38-6 和图 38-7 所示。一般原始信号与重建信号之间差值的范数除以原始信号的范数，在这种情况下，重建的误差比较小，重建信号保留了原始信号绝大部分的能量。

（4）函数 Hilbert 计算实数输入序列 x 的 Hilbert 变换，返回一个同样长度的复数结果：

```
y=Hilbert(x)
```

其中 y 的实部就是原始的实信号，虚部是实际的 Hilbert 变换，y 有时被称为解析信号。离散解析信号的主要特性是其在单位圆的下半部分的 z 变换为零，而解析信号的许多运用与此有关。Hilbert 变换与实际信号之间有 90 度的相移，如下面的例子，比较原信号与经过 Hilbert 变换后的图形：

```
t=0:1/1023:1;
x=sin(2*pi*60*t);
y=Hilbert(x);
plot(t(1:50),real(y(1:50))),hold on
plot(t(1:50),imag(y(1:50)),':'),hold off
```

所得图形如图 38-8 所示，其中实线为原信号图，虚线为经过 Hilbert 变换的图形。

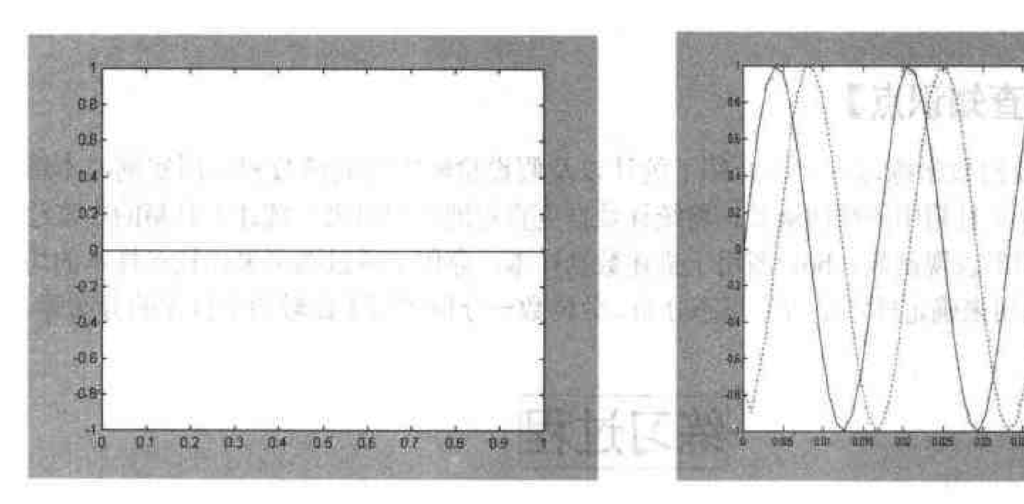


图 38-7　经过 DCT 变换的信号图　　　　图 38-8　Hilbert 信号变换图

【练习小结】

本练习主要介绍了信号变换的基本函数，包括离散傅立叶变换、快速离散傅立叶变换、Chrip-z(CZT)变换、离散因果变换和 Hilbert 变换等，并通过一些实例来说明了这些函数的用法。

【思考题】

了解信号变换的几种方式和其基本思想，复习这些信号变换在 MATLAB 中的实现以及实现的函数，在 MATLAB 中实现本练习中介绍的实例。

练习 39 统 计 绘 图

数学知识背景

在统计分析以及统计计算中，统计绘图可以直观地描绘数据的特征，更直接地勾画出统计数据的变化趋势。

主要内容

【本练习考查知识点】

由于在先前的数学练习中已经介绍了统计以及假设检验方面的函数和运用实例，本练习着重介绍统计工具箱中的绘图函数，将统计数据更直观地展现出来。统计工具箱的丰富绘图功能增添了图形表现函数，box 图用于描述数据样本，亦用于通过图形来比较多样本的均值，正态概率图用来确定样本是否为正态分布，分位数一分位图用于比较两个样本的分布等。

练习过程

（1）box 图主要用来进行图形化的检验，进行数据样本的 box 图，其函数为“boxplot”，用法是：

```
boxplot(X,notch, 'sym',vert,whis)
```

其中，X 为矩阵，X 中的每列数据绘制一个 box 图。Notch 缺省则 box 图无切口，取值为 1 时，图形带切口。“sym”为野值标记，较省符号为“+”。Vert 控制 box 图水平或者垂直放置，取值为 0 时，水平放置；取值为 1（vert=1）时垂直较置。Whis 定义虚较的长度为内四分位间距（IRQ）的函数，缺省为 1.5*IRQ，取值为 0 时，则 box 图用 sym 规定的标记数据。例：

```
a=normrnd(4,1,100,1);b=normrnd(5,1,100,1);
x=[a b];
boxplot(x,1)
```

其 box 图如图 39-1 所示，由图可见 X 中两列数据均值之较约差 1。

（2）误差条图的主要函数为“errorbar”，其用法为 errorbar（x,y,l,u,symbol），功能是给

出 x-y 图及由 l 和 u 规定误差界限的误差条，symbol 为一个字符串，可规定类型和颜色。例

```
a=0.1:0.2:0.5;r=poissrnd(a(ones(50,1),:));
[p,pci]=poissfit(r,0.001);
l=p—pci(1,:);
u=pci(2,:)—p;
errorbar(1:3,p,l,u, 'sym');
```

所得结果略。

（3）最小二乘数据可以将拟合前数据与拟合后所得直线和曲线进行绘图比较，这里有直接进行数据最小二乘拟合线的函数 lsline，例：

```
a=[1 2.2 3.2 4.1 5.3 6 8.4 10.2 13.1 15.3];
plot(a,'+');
lsline
```

在命令窗口中运行得到结果如图 39-2 所示。

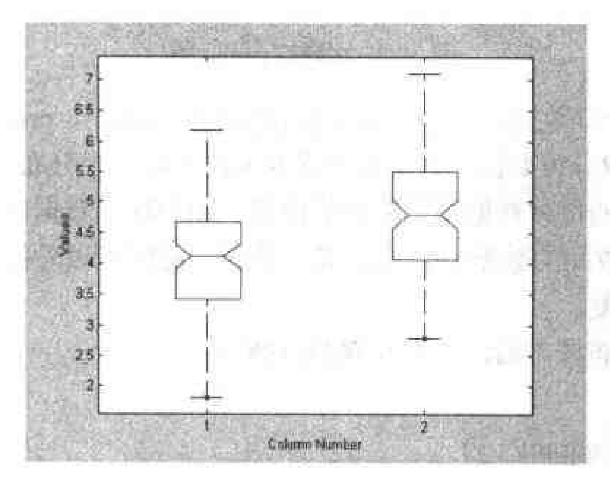


图 39-1　box 图

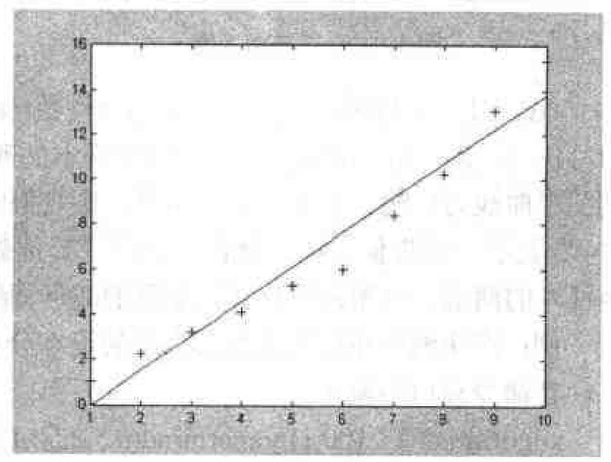


图 39-2　数据最小二乘拟合线

（4）正态数据图可用来绘制图形化的正态检验正态概率图，其函数为 normplot，其用法为：normplot(x)。其中图形绘制的是数据 x 的正态概率图，当 x 为矩阵时，图形为 x 的每一列绘制一条线，图形用符号“+”标识样本数据。当数据分布服从正态分布，则图形呈现为直线，而其他概率函数则表现出不同程度的弯曲，结果如图 39-3 所示。

```
x=normrnd(0,1,80,1);
normplot(x)
```

（5）绘制数据排列的函数有“pareto”，其用法为 pareto(y,'names')，其中 names 是可选的，而 y 则为一列数据。函数将数据 y 按数据递减顺序绘制成直方图，其上的折线则表示累积频率。而 pareto（y,'names'）则用字符串 names 中的名称对数据 x 中对应元素进行标记。假设有若干中水果品种，并且相应品种对应一定数量的水果，我们用 pareto 图进行绘制：

```
fruits=['apples ';'pears   ';'bananas';'oranges'];
amounts=[7,12,24,16];
pareto(amounts, fruits)
```

注意应该使数组中的元素具有相同长度或大小，当数组中各个元素长度不符时，系统将

给出错误信息：

All rows in the bracketed expression must have the same number of columns.

所绘图形如图 39-4 所示。

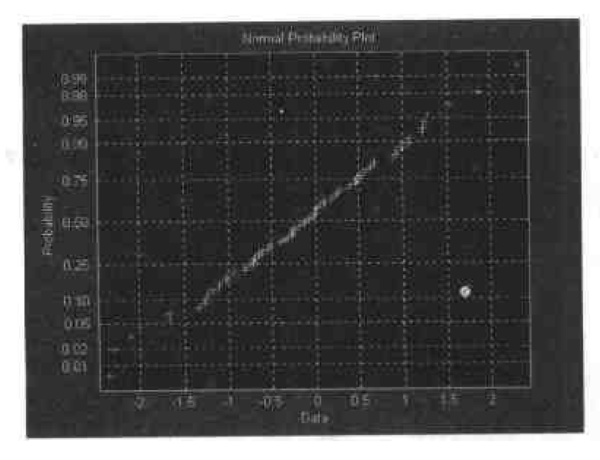


图 39-3　正态概率图

图 39-4　水果的 pareto 图

（6）双分位数图绘制两个样本的分位数-分位数图，其函数主要为 qqplot(x,y,pvec)，prev 为可选项，为矢量。函数主要显示两个样本的双分位数图。如果两个样本来源于同一个分布，则图中曲线为直线。若 x 和 y 为矩阵，则它们的每列数据绘制单独的曲线。图中样本数据以“+”表示。并将位于第一分位数和第三分位数间的数据拟合绘制成一条线。此线可以外推至样本的两端，以帮助用户评估数据的线形程度。

例：产生两个不同均值和标准差的正态分布样本后，作出其双分位数图。

在命令窗口中输入：

```
x=normrnd(0,1,100,1);y=normrnd(0.5,2,50,1);qqplot(x,y)
```

运行后得到结果为如图 39-5 所示。

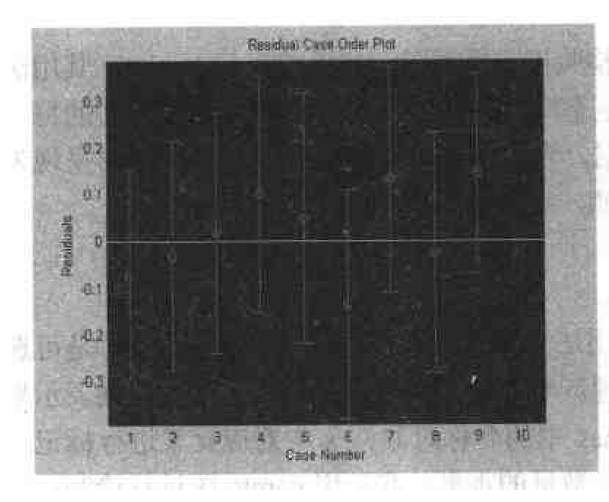


图 39-5　正态样本的双分位数图

图 39-6　回归参差图

（7）回归参差图的主要函数是 rcoplot，其用法为 rcoplot(r,rint)，函数将实验样本回归后的残差及其置信区间绘制成误差条图，其中参数 r 和 rint 来自函数 regress 的输出。

例：

```
x=[ones(10,1) (1:10)'];y=x*[10;1]+normrnd(0,0.1,10,1);
[b,bint,r,rint]=regress(y,x,0.05);
rcoplot(r,rint)
```

运行可得到结果如图 39-6 所示。

（8）对于多项式的图形我们可以使用函数 refcurve(p)，其中 p 是多项式中以系数表示的数组向量。

例：已知多项式：$s=f(x)=-x^2+3x-1$，又知测定所得的一组数据：

s=[－1.1 0.9 1.9 －0.9 －5.4 －10.5 －19.6 －28.7 －40 －55.5 －71.6]，试对实际数据和理论公式进行比较。

在命令区中运行：

```
s=[－1.1 0.9 1.9 －0.9 －5.4 －10.5 －19.6 －28.7 －40 －55.5 －71.6];
plot(s,'+')
refcurve([－1 3 －1])
```

运行后可得到结果如图 39-7 所示曲线。

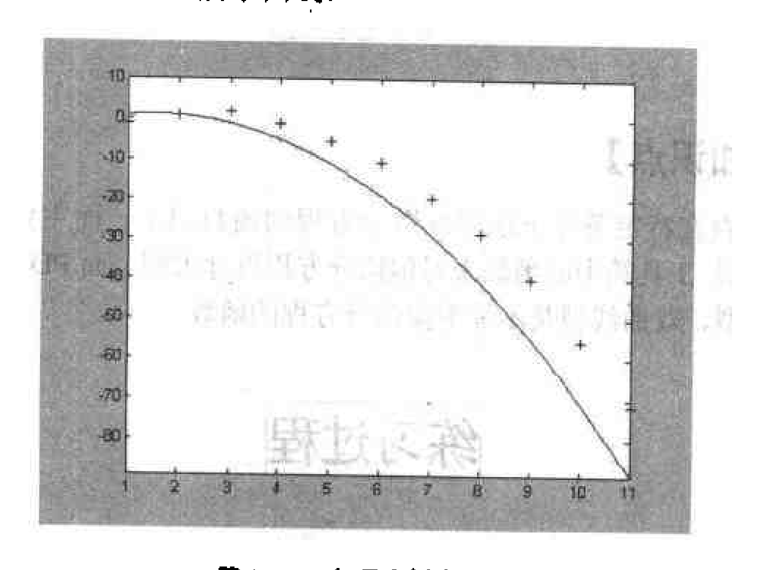


图 39-7　多项式拟合曲线

【练习小结】

本练习主要介绍了统计工具箱中进行统计绘图的函数，通过绘制统计数据的图形和相关统计参数来更形象地考察研究对象。

【思考题】

1．复习本练习介绍的绘制统计数据和其相关参数的函数，并在 MATLAB 工作区中实现。

2．对有关概率论与数理统计中的数据进行研究绘图。

练习 40　偏微分方程工具箱

数学知识背景

与解常微分方程一样，求解偏微分方程只有在一定条件下，才能得出其解析解。在工程数学计算中，经常要求解偏微分方程。我们可以在一定初始条件和特殊情况下得到偏微分的数值解，对于一定形式的偏微分方程，MATLAB 提供了求解对应偏微分方程的工具包和相应函数。

主要内容

【本练习讲述知识点】

本练习主要考查在特定条件下求解偏微分方程的函数用法，而主要运用 PDE（Partial Differential Equation）工具箱中的函数来对偏微分方程进行求解。而 PDE 工具箱中，则主要提供了求解抛物线型、双曲线型及本征型偏微分方程的函数。

练习过程

（1）求解抛物线型的函数主要是 parabolic，此函数主要是用有限元法求解抛物线型微分方程以及微分方程组，其用法为：

u1=parabolic(u0,tlist,b,p,e,t,c,a,f,d)

网格参数主要为 p、e 和 t，边界条件 b 可以用矩阵形式也可以用 m 文件格式，主要依赖于时间 t，方程系数 c、a、d、f 也可是时间的函数。tlist 是时间序列，可以在函数末加入误差限来控制。对于标量形式的偏微分方程，函数返回值为一个矩阵。

u1=parabolic(u0,tlist,K,F,B,ud,M)

这个函数主要用于求解 ODE 问题，其中初值为 u0。

例：求热传导方程

$$\frac{\partial u}{\partial t} = \Delta u$$

求解的范围为正方形区域：$-1 \le x, y \le 1$，

初值条件：当 $x^2 + y^2 \le 1$ 时，u(0)=1，在其他条件下，u(0)=0。

在命令区中键入下命令得：

```
[p,e,t]=initmesh('squareg');
[p,e,t]=refinemesh('squareg',p,e,t);
u0=zeros(size(p,2),1);
ix=find(sqrt(p(1,:).^2+p(2,:).^2)<0.4);
u0(ix)=ones(size(ix));
tlist=linspace(0,0.1,20);
u1=parabolic(u0,tlist,'squareb1',p,e,t,1,0,1,1);
```

求得的结果为：

```
Time: 0.00526316
Time: 0.0105263
Time: 0.0157895
Time: 0.0210526
Time: 0.0263158
Time: 0.0315789
Time: 0.0368421
Time: 0.0421053
Time: 0.0473684
Time: 0.0526316
Time: 0.0578947
Time: 0.0631579
Time: 0.0684211
Time: 0.0736842
Time: 0.0789474
Time: 0.0842105
Time: 0.0894737
Time: 0.0947368
Time: 0.1
96 successful steps
0 failed attempts
194 function evaluations
1 partial derivatives
20 LU decompositions
193 solutious of linear systems
```

图 40-1　求解偏微分方程运行图

即经过了 96 步计算，失败次数为 0，194 次的函数赋值，1 次求偏导，20 次求 LU 分解

运算，193 组线性系统的解。

（2）求解双曲线型偏微分方程的主要函数是 hyperbolic，它也是用有限元的方法来求解偏微分方程或者方程组，其用法为：

u1=hyperbolic(u0,ut0,tlist,b,p,e,t,c,a,f,d)

u0 是初始值，网格参数为 b、e、t，边界条件 b 可用矩阵形式也可以用 m 文件格式，它依赖于时间，而方程系数 c、a、d、f 也可以是时间的函数，可以对函数设置误差限。

u1=hyperbolic(u0,ut0,tlist,K,F,B,ud,M)

可以用于求解 ODE 问题，初值为 u0。

（3）求解特征值问题主要用函数 pdeeig，函数主要用有限元法来求解特征方程，其用法为：

[v,1]=pdeeig(b,p,e,t,c,a,d,r)

参数 p、e、t 描述区域，参数 b 描述边界条件，r 两个元素的数组。所得结果中，v 是特征向量矩阵，对于标量形式的特征值方程，v 对应于网格节点的特征值。

求解一般稀疏特征值问题的函数主要是 sptarn，其主要用法是：

[xv,lmb,iresult]=sptarn(a,b,lb,ub,spd,tolconv,jmax,maxmul)

函数主要对特征多项式（$(A-\lambda B)x=0$ 在区间[lb,ub]上的特征值，A 和 B 是稀疏矩阵，lb 和 ub 分别是要求解的特征值的上界与下界。若要求解的是 ub 左边的所有特征值，则可以令 lb=－inf，若要求解的是 lb 右边的所有特征值，则令 ub=inf。对于窄区间，可以较快得到结果。在复数情况下，比较 lmb、lb、ub 的实数部分。xv 是特征向量，它的值使得判断式(a*xv－b*xv*diag(lmb))最小。lmb 是对应的特征值，当 iresult 大于或者等于 0 时，可以进行求解并找到所有特征值，而当 iresult 小于 0 时，则求解可能是不完全的，有可能有特征值未求出。如果特征值均为正值，则 spd=1，tolconv 是期望的相对精度，缺省值为 100*eps，这里 eps 为机器特度。jmax 是基向量的最大数目，求解中需要 jmax*n 的工作空间。Maxmul 是 Arnoldi 运行次数，应是所有特征值的倍数。

例：求解方程：$-\nabla u=\lambda u$

在 L 型区域上的小于 100 的特征值及其相应的特征模态，命令窗口中演示有：

```
[p,e,t]=initmesh('lshapeg');
[p,e,t]=refinemesh('lshapeg',p,e,t);
[v,l]=pdeeig('lshapeb',p,e,t,1,0,1,[-Inf 100]);
pdesurf(p,t,v(:,16))
```

运行结果为：

```
              Basis= 10,  Time=  0.65,  New conv eig=  0
              Basis= 19,  Time=  1.09,  New conv eig=  3
              Basis= 28,  Time=  1.59,  New conv eig=  6
              Basis= 37,  Time=  2.25,  New conv eig=  9
              Basis= 46,  Time=  3.07,  New conv eig= 13
              Basis= 55,  Time=  4.01,  New conv eig= 27
End of sweep: Basis= 55,  Time=  4.01,  New conv eig= 27
              Basis= 37,  Time=  4.77,  New conv eig=  0
```

```
                Basis= 46,   Time=     5.54,   New conv eig=   0
End of sweep :  Basis= 46,   Time=     5.54,   New conv eig=   0
```

（4）求解非线性偏微分方程可以用函数 pdenonlin，其用法为：

```
[u,res]=pdenonlin(b,p,e,t,c,a,f)
```

这个函数主要用来求解非线性标量形式的偏微分方程，其中，c、a、f 是依赖于 u 的函数，该函数主要用牛顿迭代法进行求解。在参数中，主要用于设置方程的迭代次数、迭代中止误差或者初解等。

例：求解最小表面积的问题，在命令框中输入：

```
g='circleg';b='circleb2';c='1./sqrt(1+ux.^2+uy.^2)';
a=0;f=0;rtol=1e－3;[p,e,t]=initmesh(g);[p,e,t]=refinemesh(g,p,e,t);u=pdenonlin(b,p,e,t,c,a,f,'tol',rtol);
pdesurf(p,t,u);
```

运行结果如图 40-3 所示。

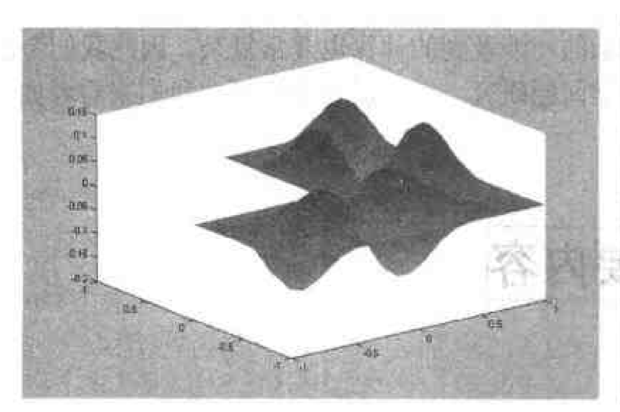


图 40-2　特征模态图

图 40-3　最小表面求解

【练习小结】

本练习介绍了求解偏微分方程特例的求解，对在一定条件下的偏微分方程，MATLAB 有对应的函数来求解．读者在看完本练习后应复习掌握各个函数的用法。

【思考题】

求解波动方程：

$\frac{\partial^2 u}{\partial^2}=\Delta u$，求解区域为 $-1\le x,y\le 1$，边界条件：$x=\pm 1$ 时，u(0)=0，且 $\frac{\partial u}{\partial n}=0$，当 $y=\pm 1$ 时，u(0)=arctan(cos(π x))，且有：$\frac{du(0)}{dt}=3\sin(\pi x)\exp(\cos(\pi y))$。

练习 41　力学基础问题

知识背景

从本练习开始，我们来学习用 MATLAB 解力学问题。力学是最为基础的学科之一，它最早来自于物理学，后来逐渐分离出来，形成单独的一门学科。力学和我们的日常生活有着密切的联系，我们身边随处可见力学现象：有常见的运动力学，也有和我们生活密切相关的材料力学。事实上，有些力学问题是相当复杂的，涉及到的计算也非常复杂。因此我们考虑采用数值计算和数据可视化的方法来完成力学问题研究。在以后 5 个练习中我们来看看如何利用 MATLAB 解决力学问题。

主要内容

【本练习讲述知识点】

本练习将要介绍用 MATLAB 解决力学中一些基本问题。将会用到 roots 求根命令，max 求极大值的命令，还要接触到进行字符串计算的命令 eval 函数。另外，我们还将简单学习如何使用 axis 语句调整坐标轴比例。

练习过程

（1）飞行问题

设目标相对于发射点的高度为，给定初速，求物体在真空中飞行的时间和距离。

这是一个无阻力抛射的问题。目标和出射点不在再一高度上。如果人工计算，其实并不难，但我们考虑建立对一类问题的解法，以后遇到类似问题，可以很容易的解决。

设初速为 V_0，抛射角为 θ_0，目标高度为 y_f，则飞行方程为：

$$y = V_0 \sin\theta_0 \cdot t - \frac{1}{2}gt^2 = y_f$$

这是一个二次方程，显然解出的 t 有两个值，应该根据实际情况进行取舍。

得出飞行时间后，我们就可以得出水平飞行距离：

$$x_{max}=V_0\cos\theta_0\cdot t_f$$

有了方程后，我们就能根据建立的模型编制程序。我们编好程序并输入命令区：

```
y0=0;x0=0;
yf=10;
V0=50;
angle=45;
V0x=V0*cos(angle*(pi/180));
V0y=V0*sin(angle*(pi/180));
wy=-9.81;wx=0;
tf=roots([wy/2,V0y,y0-yf]);
tf=max(tf);
t=0:0.001:tf;
y=y0+V0y*t+wy*t.^2/2;
x=x0+V0x.*t+wx*t.^2/2;
xf=max(x),plot(x,y)
gtext('x')
gtext('y')
```

将得到图 41-1 和 xf 的值：

```
xf =
  244.4115
```

从图上我们可以清楚地看出抛体运动的轨迹（请注意坐标间隔取值不同）。xf 的值就是我们要求的抛体的水平运动距离，同时求得 tf=6.9131s。

我们改变参数看一下图形有什么不同。取 yf=500，V0=100，angle=30，绘制图形，得到图 41-2 和新的 xf 值：

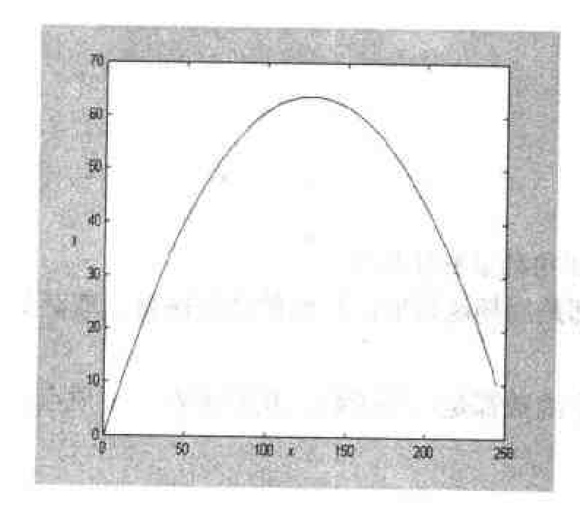


图 41-1　抛体飞行轨迹

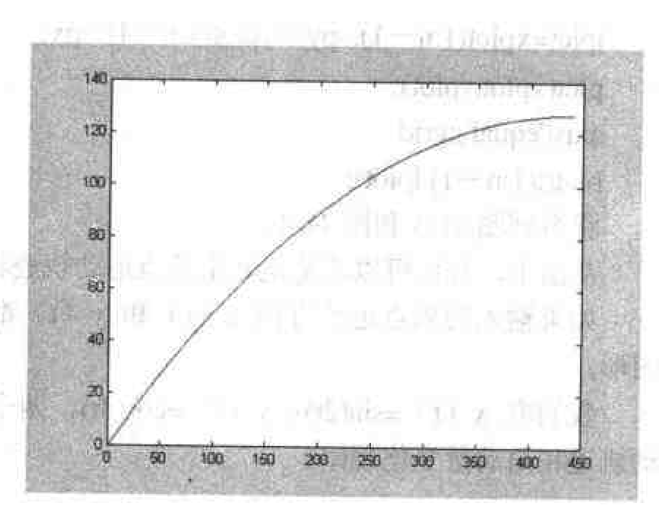


图 41-2　改变参数后的飞行轨迹

```
xf =
  441.3265
```

```
tf =
   5.0968 + 8.7154i
```

请读者比较图 41-1 与图 41-2，自己分析为什么轨迹不同？再看一下 tf 的值，发现 tf 是复数。这说明抛体根本到不了目标高度，也就是说，抛体不会击中目标。

力学问题不同于数学问题，有着自身的特点，必须根据实际情况，分析确定结果的合理性，否则会得出非常荒谬的结论。

（2）轨迹问题

已知质点沿 x 轴和 y 轴方向的运动规律是 x（t）和 y（t），求运动轨迹和对于原点的角动量。

我们来分析这个问题：

设角动量为 $\vec{L}$，质点的动量为 $\vec{P}=m\vec{v}$，矢径为 $\vec{r}$，有

$$\vec{L}=\vec{r}\times\vec{P}=\vec{r}\times m\vec{v}$$

在平面上就是：

$$L=x\cdot mv_y-y\cdot mv_x$$

下面我们就编制程序，并输入命令区：

```
x='t.*cos(t)';
y='t.*sin(t)';
tf=20;
m=1;
n=100;t=linspace(0,tf,n);dt=tf/(n－1);
xplot=eval(x);yplot=eval(y);
px=m*diff(xplot)/dt;
py=m*diff(yplot)/dt;
lplot=xplot(1:n－1).*py－yplot(1:n－1).*px;
plot(xplot,yplot);
axis('equal');grid
plot(t(1:n－1),lplot);
```

将得到图 41-3 和图 41-4。

从图上，我们可以清楚地看出质点运动轨迹和角动量变化规律。

如果输入的质点运动方程 x（t）和 y（t）都是周期函数时，得到的轨迹图就是李萨如图形。

我们取 x（t）=sin(2t)；y（t）=cos(3t)，两个函数都是周期函数。我们来看一下质点运动轨迹和角动量变化规律。

图 41-3　运动轨迹

图 41-4　角动量与时间关系

将运动方程输入命令区后（程序略），将得到图 41-5 和图 41-6。

图 41-5 所示的轨迹就是一个李萨如图形，请读者对比与图 41-3 之间的区别。

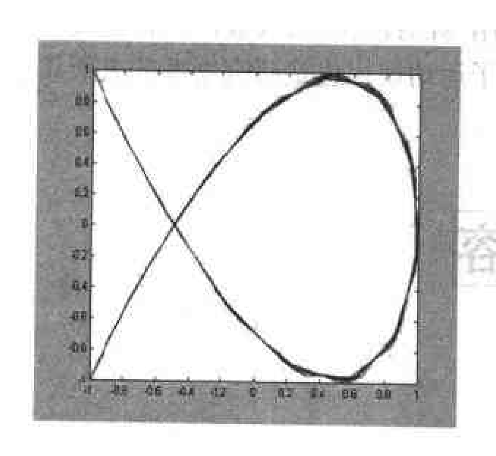


图 41-5　运动轨迹图

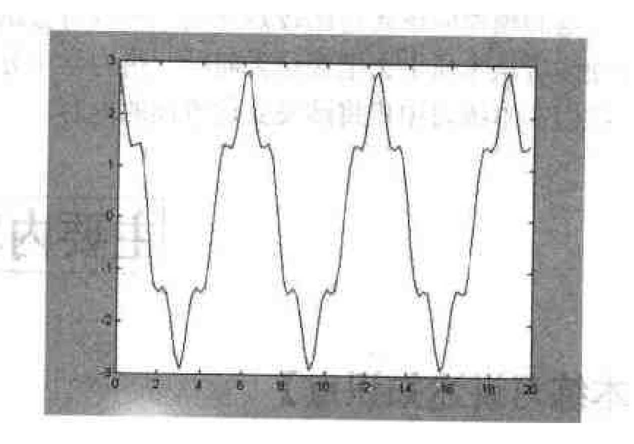


图 41-6　角动量与时间关系

【练习小结】

本练习作为力学部分的第一个练习，旨在引导读者对力学问题有一个初步了解。我们学习用 MATLAB 作为工具解决具体力学问题。在求解过程中，我们用到了多项式求根的命令 roots 语句，并且在不能确定结果取舍时使用 max 或 min 命令求得合适的解。axis ('equal') 用来调整横纵坐标有相同的比例。eval 是进行字符串计算的函数。请读者注意 eval 函数在本练习中的应用。diff 函数用来作近似的导数计算。本练习知识点较多，请仔细揣摩，认真记忆。

【思考题】

1. 请回忆 roots 函数命令的用法，它是用来解什么类型多项式的？
2. 练习中的复数解说明什么问题？能否从图形中看出解的不合理性？
3. 请用 axis 命令调整图 41-1 比例，使轨迹以相同坐标间隔表示出来。
4. diff 函数的作用是什么？

练习 42　碰撞和热力学

知识背景

碰撞是在力学中经常遇到的一类问题，求解这类问题一般要用到动量守恒、能量守恒定律以及其他物理规律。相信很多读者在中学时就经常与小球碰撞之类的问题打交道，对于这类问题的模型应该还是比较熟悉的。本练习尝试用 MATLAB 来求解，并将结果可视化，从而使读者从本质上去把握读类问题。热力学中分子运动规律也是研究物理的人所乐意探讨的，我们在本练习中也将涉及到这方面的内容。

主要内容

【本练习讲述知识点】

本练习首先介绍小球的弹性碰撞问题，训练读者将实际问题转化为数学模型并进而用 MATLAB 语句加以实现的能力。我们还将介绍热力学中分子速度的求解问题。本练习中用到了 roots 命令，在编程中使用了限制性语句。请读者仔细体会这些内容。

练习过程

（1）小球碰撞 v_0

我们先来看一个中学时很熟悉的题目。

有一个质量为 m 的小球以速度 u_0 正面撞击质量为 M 的静止小球，若碰撞是完全弹性碰撞，求碰撞后两球的速度，并讨论速度与两球质量比 M/m 之间的函数关系。

我们来分析这个问题。完全弹性碰撞即无能量损失的碰撞，这大大方便了问题的讨论。我们假设碰撞后两球速度与 u_0 同向，球 m 的速度为 u，M 球的速度为 v，设质量比为 k=M/m，取 $v_r=v/u_0$，$u_r=u/u_0$，列出动量守恒和能量守恒方程式如下：

$m u_0 = m u + Mv$　　　　　　动量守恒

$$\frac{1}{2}mu_0^{\ 2} = \frac{1}{2}mu^2 + \frac{1}{2}Mv^2$$　　　　动能守恒

对这两个式子进行变换，化为下面的形式：

$$K u_r + v_r = 1;$$
$$K u_r^2 + v_r^2 = 1;$$

整理后，得到 $(1-u_r)^2 + K u_r^2 = K$

m 球的能量损失为 $E_m = \frac{1}{2}m(u_0^2 - u^2) = \frac{1}{2}mu_0^2(1-u_r^2)$

整理后，得到 $(1+\frac{1}{k})u_r^2 - \frac{2}{k}u_r + (\frac{1}{k}-1) = 0$

这是一个一元二次方程，我们可以很容易地求出 u_r。编制程序，并输入命令区：

```
K=logspace(-1,1,11)
for i=1:length(K)
u1=roots([(1+1/K(i)), -2/K(i),(1/K(i)-1)]);
u(i)=u1(abs(u1-1)>.001);
end
v=(1-u)./K;
de=1-u.*u
u, v
semilogx(K,[u;v;de]),grid
gtext('de')
gtext('v')
gtext('u')
```

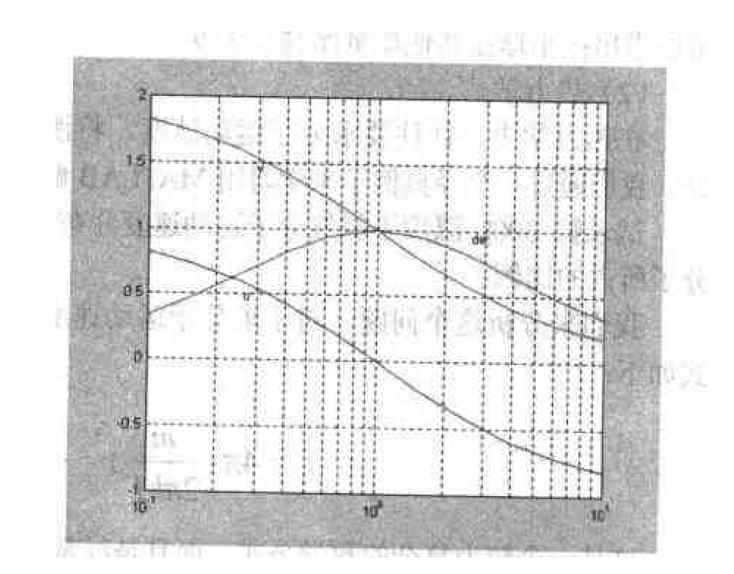


图 42-1 球速与 K 值关系曲线

得到的图形如图 42-1 所示（u_r 即图中的 u，v_r 即图中的 v）。

我们来分析图线的意义。对于 m 球相对动能损失 de 来说，随着 K 值增大，de 先增大，后变小。当 K 值为 1 时，de 达到最大值，即全部动能都传给了 M 球。对于 m 球相对速度 u 来说，随 K 值增大，一直减小。K=1 时，u=0；K<1 时，u<0。对于 v，则随着 K 值增大，v 持续减小，K=1 时，即 M=m 时，M 球速度等于 u。

我们再来看一下得到的计算结果：

```
K =
   Columns 1 through 7
    0.1000    0.1585    0.2512    0.3981    0.6310    1.0000    1.5849
   Columns 8 through 11
    2.5119    3.9811    6.3096   10.0000
de =
   Columns 1 through 7
```

```
    0.3306    0.4724    0.6418    0.8147    0.9488    1.0000    0.9488
  Columns 8 through 11
    0.8147    0.6418    0.4724    0.3306
u =
  Columns 1 through 7
    0.8182    0.7264    0.5985    0.4305    0.2263         0   -0.2263
  Columns 8 through 11
   -0.4305   -0.5985   -0.7264   -0.8182
v =
  Columns 1 through 7
    1.8182    1.7264    1.5985    1.4305    1.2263    1.0000    0.7737
  Columns 8 through 11
    0.5695    0.4015    0.2736    0.1818
```

从计算结果中，我们可以很清楚地看出变化规律。利用所得数据进行插值计算，我们可以求出在小球在其他质量比时的速度。

（2）热力学

在热力学中，往往要求分子运动速度。将涉及到指数运算，用人工方法将显得比较繁杂。我们通过一个经典例子来学习用 MATLAB 解热力学问题。

试求解 300k 温度下氮气分子运动速度分布曲线，并求速度在（300，500）m/s 范围的分子所占的比例。

我们来分析这个问题。对于求分子运动速度的问题，要用到麦克斯韦速度分布率。形式如下：

$$f = 4\pi\left(\frac{m}{2\pi kT}\right)^{3/2} \cdot v^2 \cdot \exp\left(-\frac{mv^2}{2kT}\right)$$

这是一个较为复杂的数学公式，而且是经常用到的公式。我们考虑编制一个函数 M 文件，这样我们便能够方便地实现调用。

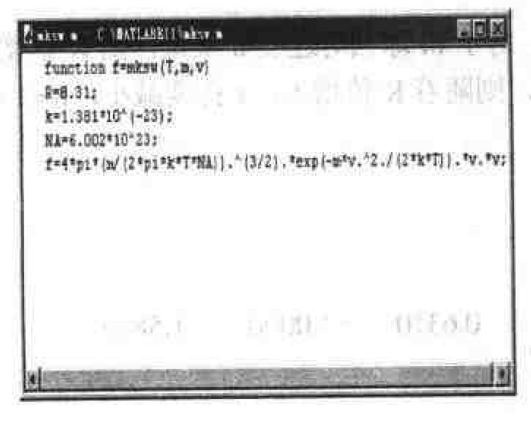


图 42-2 mksw.m 文件

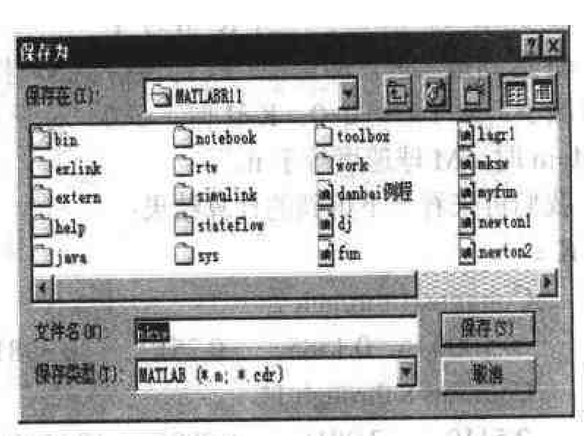


图 42-3 保存函数 M 文件

编制 M 文件如图 42-2 所示。

将文件命名为 mksw.m ，并保存在 C:\MATLABR11，见图 42-3。并且在 MATLAB 路径浏览器里将路径设置为 C:\MATLABR11。

我们在命令区里输入：

```
T=300;
m=28e-3;
v=0:1500;
y=mksw(T,m,v);
trapz(y1)
```

将得到下面的结果：

```
ans =
     0.3763
```

结果表明速度在（300，500）m/s 范围的分子占所有分子的比例为 0.3763。

【练习小结】

本练习主要向大家介绍了用 MATLAB 解决热力学问题的方法。我们在编程过程中，用到了 roots 命令求解多项式的根；用到了限制语句 u(i)=u1(abs(u1－1)>.001)，限制语句将按照条件对数值进行取舍。另外，我们复习了 M 文件的编写、保存和调用，在一个较为复杂的程序里，我们要善于嵌入 M 文件，这样将使主程序简洁、明了。

【思考题】

1．在小球碰撞的例子里，我们是在完全弹性碰撞的条件下求解的，如果不是完全弹性碰撞，应该怎样建立模型？

2．在小球碰撞的程序中，如果不用限制性语句，能否完成运算？你还有别的办法吗？

3．图 42-1 为什么要用半对数坐标绘图？

4．调用 M 文件应该注意哪些问题？如果没设置好路径，将出现什么结果？

练习 43 振　　动

知识背景

振动是很常见的现象。平常我们看到树叶被风吹动，乒乓球在地上反弹，都是振动现象。还有我们平常听到的声音，是振动物体引起空气振动，被耳朵感知的结果。我们所处的世界是一个随处可见都动的世界，如果没有振动，我们所处的环境将变得死寂而没有生机。振动现象有着它自身的特点，了解并能解释简单振动现象是提高我们对自然界认识度的必要条件。

主要内容

【本练习讲述知识点】

本练习首先介绍了一般性振动问题的解法，两个例子都是比较经典的问题。求解过程中，要用到 pause、sound 命令。这两个命令我们接触不多，希望读者能够好好体会它们的用法。mesh 命令用来绘制三维图形；atan2 是都一次接触，它用来求四个象限的反正切。

练习过程

（1）振动合成和拍频

事实上，我们经常见到的振动现象是若干个简单振动的叠加。比如水中的波纹，就是由若干个振动叠加的结果。下面，我们就来研究如何叠加两个振动。有两个同方向的振动：

$y_1 = a_1\sin(\omega_1 t+\varphi_1)$ 和 $y_2 = a_2\sin(\omega_2 t+\varphi_2)$

叠加，有 $y = y_1 + y_2 = a_1\sin(\omega_1 t+\varphi_1)$ + $\sin(\omega_2 t+\varphi_2)$ 进一步得

$$y=(a_1+a_2)\sin\left(\frac{\omega_1+\omega_2}{2}t+\frac{\varphi_1+\varphi_2}{2}t\right)\cos\left(\frac{\omega_1-\omega_2}{2}t+\frac{\varphi_1-\varphi_2}{2}t\right)+$$

$$(a_1-a_2)\sin(\frac{\omega_1-\omega_2}{2}t+\frac{\varphi_1-\varphi_2}{2}t)\cos(\frac{\omega_1+\omega_2}{2}t+\frac{\varphi_1+\varphi_2}{2}t)$$

当 ω_1 和 ω_2 非常接近时，我们称 $\frac{\omega_1-\omega_2}{2}$ 为拍频。下面我们来叠加两个具体的振动表达式。

在命令区里输入：

```
t=0:0.001:10;
a1=5;w1=300;
a2=10;w2=310;
y1=a1*sin(w1*t);
y2=a2*sin(w2*t);
y=y1+y2;
subplot(3,1,1),plot(t,y1),ylabel('y1');
subplot(3,1,2),plot(t,y2),ylabel('y2');
subplot(3,1,3),plot(t,y),ylabel('y'),xlabel('t');
pause,sound(y1);pause(2),
sound(y2);pause(2),sound(y),pause
```

得到的图形如图 43-1 所示。

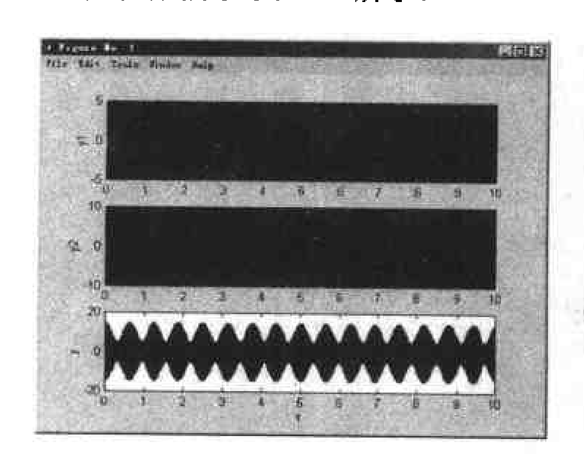


图 43-1　拍频现象

图 43-2　拍频现象不明显

图 43-1 清楚地向我们展现出了叠加前和叠加后的振动情况。从图中，我们能够看到拍频现象。

另外，在运行程序后，我们能够听到三次汽笛声。前两个分别是两个正弦波的声音，后一个是产生拍频现象的叠加波的声音。

我们来看一下程序中的 pause 命令。pause 命令有两种形式：

pause（n）　　等待 n 秒后，继续执行后续命令。

pause　　　　等待用户键入任意键后，继续执行后续命令。

pause 命令主要应用于程序调试，当程序执行到 pause 命令处时将暂停，我们可以检查

pause 命令前的程序是否有错。这是一个相当有用的方法，请读者熟记。

程序中还用到了 sound 命令，它用来产生声音。

如果我们将两个正弦波的频率取得稍远，比如 w1=300，w2=350，运行程序，将得到图 43-2。这时，拍频现象就很不明显。

同时，我们听到的第三次汽笛声也不像前面那样有明显的拍频。

（2）阻尼振动

我们来分析单自由度阻尼系统的阻尼系波对固有振动模态的影响。

单自由度阻尼系统的频动方程如下：

$$m\ddot{x}+c\dot{x}+kx=0$$

解得： $$x(t)=Ae^{-\xi\omega_n t}\sin(\omega_d t+\phi)$$

其中， $A=\sqrt{\dfrac{(v_0+\xi\omega_0 x_0)^2+(x_0\omega_d)^2}{\omega_d{}^2}}$ ， $\varphi=\arctan(\dfrac{x_0\omega_d}{v_0+\xi\omega_n x_0})$ ，

$\omega_d=\omega_n\sqrt{1-\xi^2}$

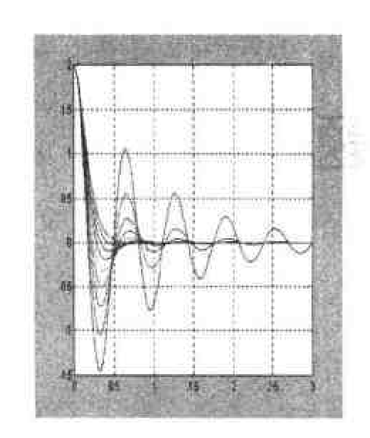
x_0 和 v_0 分别为初始位置和初始速度。

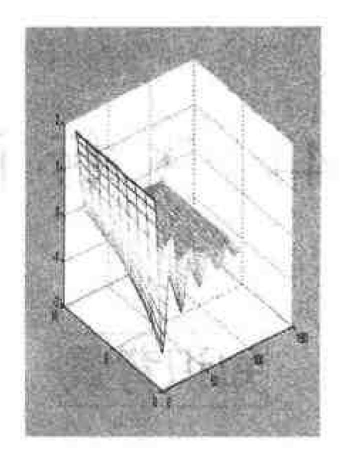
取 $\omega_n=10$， $x_0=2$， $v_0=1$，计算终点时间 $tf=3$， ξ 从 0.1 到 1。

在命令区里输入：

```
wn=10;tf=3;x0=2;v0=1;
for j=1:10
z(j)=0.1*j;
wd(j)=wn*sqrt(1-z(j)^2);
a=sqrt((wn*x0*z(j)+v0)^2+(x0*wd(j))^2)/wd(j);
p=atan2(wd(j)*x0,v0+z(j)*wn*x0);
t=0:tf/100:tf;
x(j,:)=a*exp(-z(j)*wn*t).*sin(wd(j)*t+p);
end
plot(t,x(1,:),t,x(2,:),t,x(3,:),t,x(4,:),t,x(4,:),t,x(5,:),t,x(6,:),
   t,x(7,:),
t,x(8,:),t,x(9,:),t,x(10,:))
grid,figure,mesh(x)
```

得到图 43-3 和图 43-4。





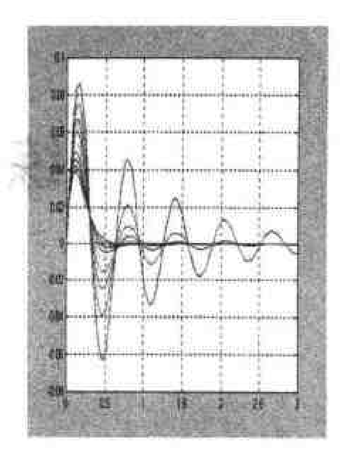


图 43-3　ζ 对固有振动模态影响　图 43-4　ζ 对固有振动模态影响（三维图）　图 43-5　ζ 对固有振动模态影响

我们可以很容易地从图中看出 ζ 对固有振动模态影响的规律。换一下参数，看一看图形有什么变化。

取 x0=0，v0=1。在命令区里输入相应命令后，将得到图 43-5 和图 43-6。

我们可以看出图形的趋势是非常不同的。可见初始条件不同造成的差别比较大。请读看自己找出取不同初值改变结果的原因。

程序中的 atan2 用来求四个象限角的反正切值。类似的有 atan 也是反正切，但它只是求某一角度的反正切值。

mesh 用来绘制三维图形，这个命令我们已经多次使用，相信读者已经比较熟悉，在这里就不再介绍了。

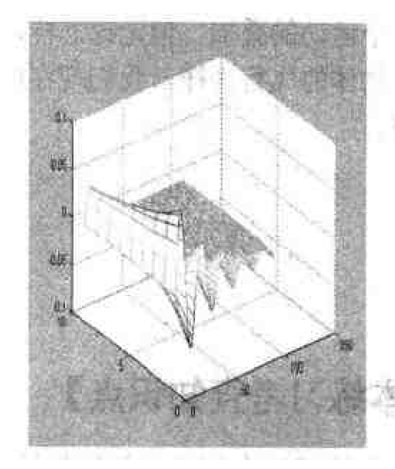


图 43-6　ζ 对固有振动模态影响（三维图）

【练习小结】

本练习主要是利用 MATLAB 来解决振动问题。pause 命令用来暂停命令执行，多用在程序调试中；sound 用来产生声音，本练习所举例子是一个很好的说明。mesh 命令用来绘制三维图形；atan2 可用来求四个象限的反正切。本练习的难点在于模型转化，如果能够将实际问题转化为模型，相信读者也能够在 MATLAB 中实现它。

【思考题】

1. 请用 pause 命令调试一个小程序。
2. sound 命令的使用格式是什么？
3. 到目前为止，学习了哪些绘图命令？
4. 能否进一步简化阻尼振动例子的程序？如何简化？

练习44 飞 行 问 题

知识背景

飞行问题也是常见的力学问题。从 18 世纪以来，人们一直对飞行问题比较重视，这是因为飞行问题与军事密切相关。世界上第一台计算机也正是因为计算导弹飞行轨迹才应运而生的。到了今天，飞行理论更是应用在航空、航天等各个领域。这些飞行通常比较复杂，是若干个运动的叠加。但无论怎样，我们都可以将问题分解，从最简单的飞行问题入手，逐步摸清物体的飞行规律。我们现在所讨论的飞行问题限于宏观世界，主要依靠牛顿经典力学来求解。

主要内容

【本练习考查知识点】

本练习将介绍飞行问题的解法。主要涉及到了高阶微分方程的解法。用到了龙格库塔函数、diff 函数和 find 语句。通过编制函数 M 文件使得主程序简洁、明了。我们还将涉及到 length 函数，请读者注意用法。

练习过程

（1）求运动轨迹和加速度

我们首先来看一个富有军事意义的问题。

设有一个导弹M，两次飞行速度分别为1000m/s和800m/s，飞行时始终对准速度为500m/s的沿直线飞行的目标 A，导弹发射点在目标运动方向的左（4000m）前（3000m）方，求导弹轨迹和加速度。

对于相对运动的问题，有时采用惯性坐标系将显得非常方便。结合题意，我们取与目标相连的等速直线运动坐标为惯性坐标系，在此坐标系里运动方程。设导弹运动速度为V_m，目标速度为V_t。则在惯性坐标系里，牵连速度为$V_e=V_t$，目标的绝对速度为$V_a=V_m$。则相对速度$V_r=V_a-V_e=V_m-V_t$，将其向 x 和 y 向投影，得到：

$$V_{rx} = -V_m \frac{x}{\sqrt{x^2+y^2}} - V_t = \frac{dx}{dt} \qquad V_{ry} = -V_m \frac{y}{\sqrt{x^2+y^2}} = \frac{dy}{dt}$$

只要对上面两个式子积分就可以求出导弹运动轨迹。

我们来编制程序。考虑到程序段会比较长，故我们先编一个函数 M 文件，然后在主程序中调用这个文件。因为要求解微分方程，所以我们采用龙格库塔方法求解。利用函数 M 文件将方程计算式储存为函数。编制的 M 文件如图 44-1 所示。

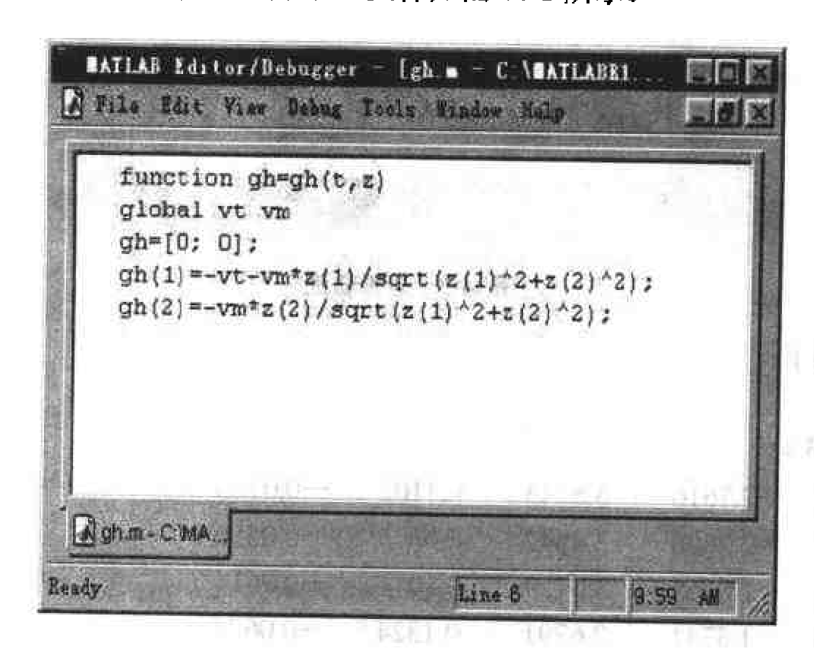


图 44-1 gh.m 函数文件

在命令区里输入主程序：

```
global vt vm
vt=500;vm=1000;
z0=[3000;4000];
tspan=[0;4.5];
[t,z]=ode23('gh',tspan,z0);
plot(z(:,1),z(:,2));grid
dt=diff(t);ddt=length(dt);
x=z(:,1);y=z(:,2);
vx=diff(z(:,1))./dt;
vy=diff(z(:,2))./dt;
wx=diff(vx)./dt(1:ddt-1);
wy=diff(vy)./dt(1:ddt-1);
[t(2:ddt),x(2:ddt),y(2:ddt),wx,wy]
```

得到的图形如图 44-2 所示。

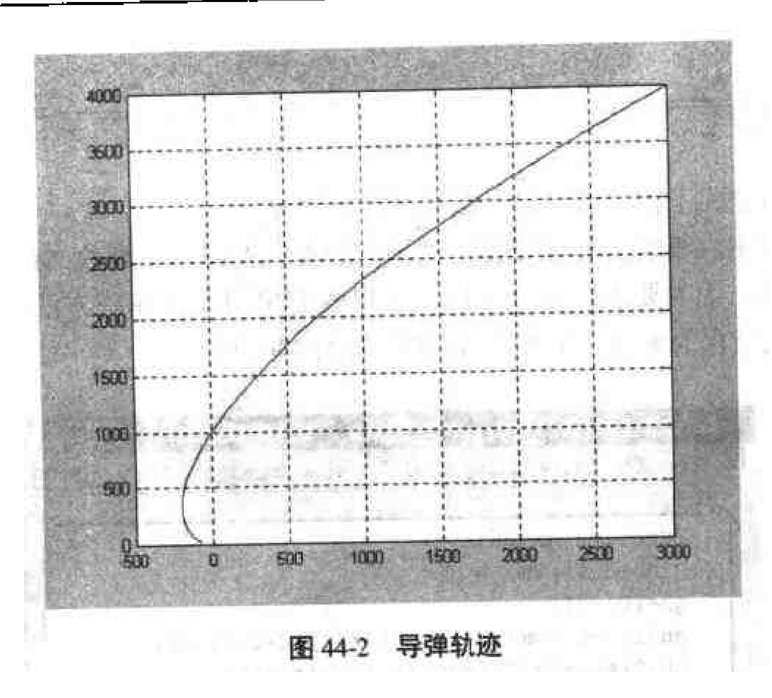


图 44-2 导弹轨迹

计算结果如下：

```
ans =
   1.0e+003 *
    0.0002    2.7616    3.8243    0.1102   -0.0783
    0.0007    2.2806    3.4542    0.0845   -0.0556
    0.0011    1.8168    3.0729    0.1044   -0.0614
    0.0016    1.3741    2.6791    0.1324   -0.0674
    0.0020    0.9582    2.2717    0.1737   -0.0723
    0.0025    0.5775    1.8496    0.2378   -0.0723
    0.0029    0.2450    1.4130    0.2461   -0.0467
    0.0032    0.0972    1.1797    0.3500   -0.0310
    0.0033    0.0076    1.0159    0.4933   -0.0022
    0.0035   -0.0686    0.8521    0.5350    0.0424
    0.0036   -0.1213    0.7135    0.7213    0.1286
    0.0038   -0.1651    0.5591    0.8004    0.2422
    0.0039   -0.1895    0.4066    0.9004    0.4255
    0.0041   -0.1900    0.2687    0.8745    0.6037
    0.0042   -0.1741    0.1787    0.9242    0.8857
    0.0043   -0.1514    0.1169    0.8842    1.1265
    0.0044   -0.1273    0.0752    0.7983    1.3271
    0.0044   -0.1045    0.0477    0.7312    1.5886
```

利用插值方法，我们可以从上面数值计算的结果中方便地查出任意时刻的导弹坐标和加速度。

（2）有阻力飞行

前面我们介绍了一个运动学方面的导弹飞行问题。现在我们再来看一个考虑阻力的动力学问题。

已知：一个质点在空中飞行，所受的空气阻力方向始终与速度方向相反，大小与速度平方成正比。求：此质点的飞行轨迹和飞行路程。

根据题意，我们可以建立如下方程（设 vx、vy 分别为质点沿 x、y 方向分速度，c 为空气阻力系数）：

$$\frac{dx}{dt}=vx \qquad \frac{dy}{dt}=vy$$

$$m\frac{dvx}{dt}=-cv^2\cos a=-cv\cdot vx \qquad m\frac{dvy}{dt}=-cv^2\sin a=-cv\cdot vy-mg$$

显然，我们又要求解高阶方程。和前面的例子一样，我们采用龙格库塔方法来求解。首先来编制一个函数 M 文件（如图 44-3 所示）。

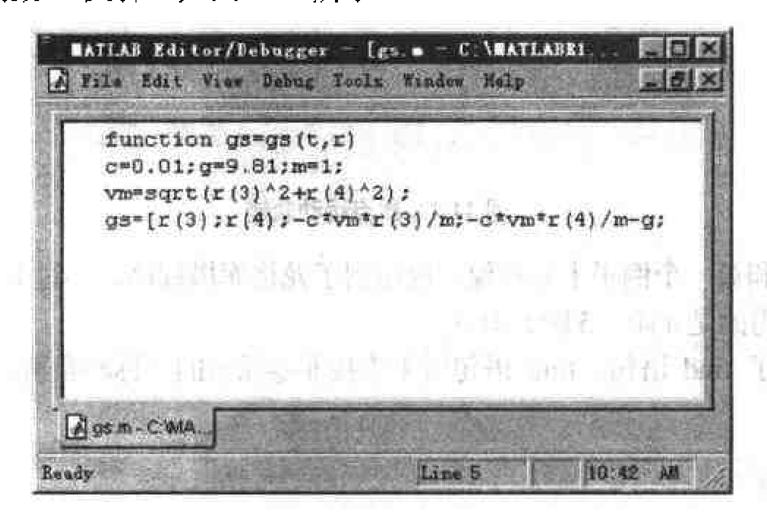


图 44-3　gs.m 函数文件

我们将文件命名为 gs.m，并将其保存在 C：\MATLABR11。同时不要忘记在路径浏览器里设置相应的路径。

我们取初始速度为 50m/s，初速度方向为 30 度，飞行时间为 6s 。在命令区里输入主程序（见下页），将得到图 44-4。

运算结果为：

xmax =

98.5357

射程为 98.5357m。

```
y0=0;x0=0;
v0=50;
angle=30;
tf=6;
vx0=v0*cos(angle*(pi/180));
vy0=v0*sin(angle*(pi/180));
r0=[0;0;vx0;vy0];
```

```
[t,r]=ode45('gs',[0,tf],r0)                   %主程序
plot(r(:,1),r(:,2)), hold on
xmax=min(r(find(r(:,2)<0),1))
plot([0,200],[0,0])
grid
```

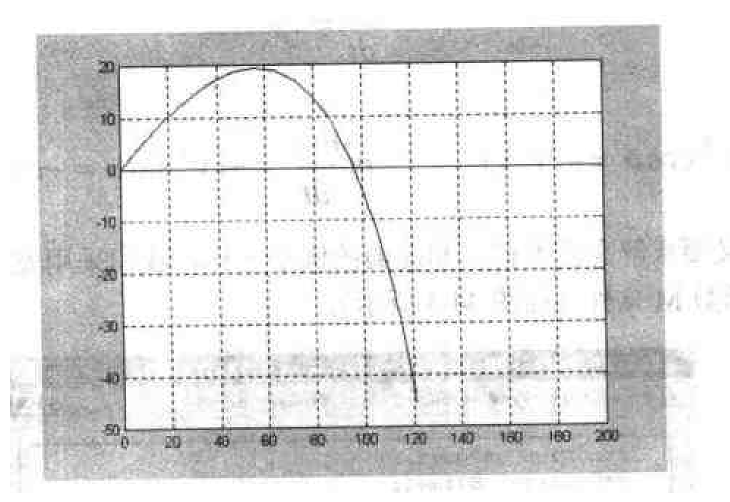


图 44-4　质点运动轨迹

本例的程序和第一个例子十分相像，也用到了龙格库塔程序，不过前例用的是 2 阶（3 阶）公式，本例用的是 4 阶（5 阶）公式。

程序中用到了 find 语句。find 语句用来查找非零元素的下标和数值。请注意 find 语句的用法。

【练习小结】

本练习主要学习了飞行问题的解法。飞行问题一般要解高阶方程，本练习使用 ode23 和 ode45 两个函数来解高阶方程。diff 函数用来于差分和求近似微分；length 用来求向量的长度。本练习的两个例子都用到了函数 M 文件。在解决同类问题时，要学会自己编制相应的函数文件，并进行调用。另外我们还用到了 find 语句，这个命令在编程时是十分有用的。本练习的程序较为灵活，请读者仔细体会程序的意义。到目前为止，我们对 M 文件已经非常熟悉了，请自己总结 M 文件的使用方法。

【思考题】

1. 一个例子中函数文件里用到了 global 命令，它的作用是什么？能否在主程序中实现这一功能？
2. ode23 和 ode45 两个函数的区别是什么？如何设置精度？
3. fiod 语句的功能是什么？使用它的好处是什么？
4. 谈谈解飞行问题的关键是什么？如何实现结果可视化？
5. 如何编辑函数 M 文件？
6. 想想如何增加程序的可读性？有哪些方法？

练习 45　材 料 力 学

知识背景

在力学问题中，还有一部分非常重要的内容：材料力学。顾名思义，材料力学的研究对象是各种材料的力学性质。在工程建设中，我们第一个考虑的问题就是建筑材料的性能。比如要建造大桥，我们就会对指定跨度下备选钢材能否承受负重进行调查。如果不合格，就不能用于建设。材料力学很大程度建立在实验的基础上，要想确切地了解一个材料的性质，必须反复实验，归纳出它的性质，进而确定它在工程上的应用。

主要内容

【本练习讲述知识点】

本练习我们将着手解决一些常见材料力学问题。我们将用到材料力学中的正应力、剪力、弯矩、挠度等概念。希望读者首先查阅相关书籍，做好知识准备。本练习将介绍 MATLAB 中一个积分函数 contrapz 的用法，请注意它的用途。

练习过程

（1）悬臂梁

在材料力学中，悬臂梁是一个非常基本的概念。有关悬臂梁的计算也很多。我们来看这样的问题。有一个悬臂梁，如图 45-1 所示。

悬臂梁长为 L，左端插入墙内，离固定端 L_1 处受集中力 P，求转角和挠度。

先来建模型。设弯矩为 M，转角为 A，挠度为 Y。取 $E=150\times10^9\text{N/m}^2$，$I=1.5\times10^{-5}\text{m}^4$，列出力学方程如下：

$$M=-P(L_1-x)\qquad(0\le x\le L_1)$$

$$=0\qquad\qquad(L_1\le x\le L)$$

$$A=\int_0^x (M/EI)dx$$

$$Y=\int_0^x Adx$$

显然需要做积分。我们使用 MATLAB 中的 cumtrapz 函数做积分。首先来简单介绍一下 cumtrapz 函数。

我们在命令区里输入：

```
Y = [0 1 2; 3 4 5]
a=cumtrapz(Y,1)
b=cumtrapz(Y,2)
```

得到的结果如图 45-2 所示。

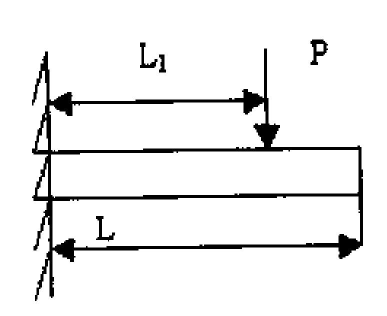


图 45-1　悬臂梁受力图

```
Y = [0 1 2; 3 4 5]
a=cumtrapz(Y,1)
b=cumtrapz(Y,2)
Y =
     0     1     2
     3     4     5
a =
          0          0          0
     1.5000     2.5000     3.5000
b =
          0     0.5000     2.0000
          0     3.5000     8.0000
```

图 45-2　cumtrapz 命令运行结果

从运行结果我们可以看出，cumtrapz（Y，1）与 cumtrapz（Y，2）的计算结果不同，请注意各自的用法。

下面我们就用 cumtrapz 函数来解上面的问题。

取 L=2m，P=1500N， L1=1m ，在命令区里输入：

```
L=2;P=1500;L1=1;
E=150e9;I=1.5e-5;
x=linspace(0,L,101);dx=L/100;
n1=L1/dx+1;
M1=-P*(L1-x(1:n1));
M2=zeros(1,101-n1);
M=[M1,M2];
A=cumtrapz(M)*dx/(E*I);
Y=cumtrapz(A)*dx;
subplot(3,1,1),plot(x,M),grid
subplot(3,1,2),plot(x,A),grid
subplot(3,1,3),plot(x,Y),grid
gtext('M')
```

```
gtext('A')
gtext('Y')
```

得到图 45-3。

从图中可以明显地看出悬臂梁弯矩、转角、挠度沿梁分布关系。在工程上，经常要绘制这样的关系图。

（2）绘制莫尔圆

已知两正交截面上的正应力σ_x、σ_y和剪应力τ_{xy}，求其它斜截面上的应力值。设斜截面与σ$_y$上的正应力σ和剪应力τ的计算公式是：

$$\sigma = \frac{\sigma_x + \sigma_y}{2} + \frac{\sigma_x - \sigma_y}{2}\cos 2\theta - \tau_{xy}\sin 2\theta$$

$$\tau = \frac{\sigma_x - \sigma_y}{2}\sin 2\theta + \tau_{xy}\cos 2\theta$$

我们在命令区里输入程序：

```
Sx=20;
Sy=0;
Txy=10;
a=linspace(0,pi,36);
Sa=(Sx+Sy)/2;Sd=(Sx－Sy)/2;
s=Sa+Sd*cos(2*a)－Txy*sin(2*a);
t=Sd*sin(2*a)+Txy*cos(2*a);
plot(s,t,Sx,Txy);
axis equal
v=axis;
line([v(1),v(2)],[0,0])
line([0,0],[v(3),v(4)])
hold,plot(Sa,0,'x')
Smax=max(s),Smin=min(s),Tmax=max(t)
```

用 gtext 命令标注后，得到图 45-4 所示的莫尔圆。

运算结果为：

```
  Smax =
          24.1101
Smin =
        －4.1386
Tmax =
        14.1101
```

若取 Sx=20，Sy=20，Txy=10，运行上面的程序，得到图 45-5。

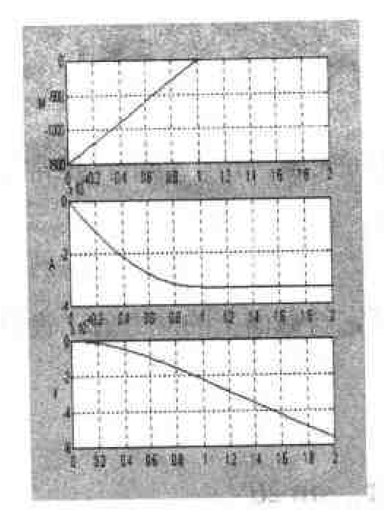


图 45-3　悬臂梁弯矩、转角、挠度曲线

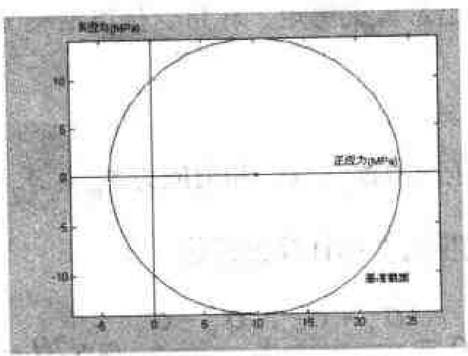


图 45-4　莫尔圆

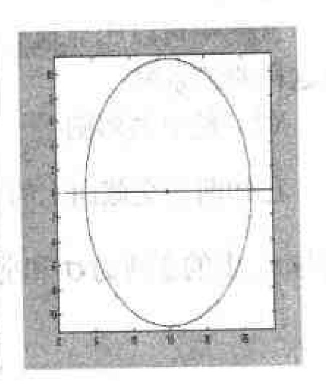


图 45-5　改变参数后的莫尔圆

得到的结果是：

Smax =

　　26.1753

Smin =

　　3.8396

Tmax =

　　11.1490

改变参数后得到正应力都为正值。

有了这个程序，输入任意参数，我们都可以求出正应力和剪应力，以后遇到这类问题，就可以用 MATLAB 轻松求解。

【练习小结】

本练习中我们遇到了材料力学中常见的问题：求悬臂梁转角和挠度；绘制莫尔圆。通过练习我们知道了 contrapz 函数的用法。contrapz 主要用来做较为精确的积分运算。在积分时，要注意灵活应用。将数据可视化始终是解决工程实际问题的目标之一，比如绘制莫尔圆，就是工程中常见任务。希望读者对照例题，仔细体会程序的意义。材料力学中的问题常常要涉及绘图，希望读者建立起以图形方法解决问题的思想，灵活运用学过的知识，以最优方法解决疑难问题。

【思考题】

1．在求解悬臂梁的程序中，如果不用 contrapz 函数进行积分，我们还可以用什么函数完成计算？

2．回忆 consum 函数的用法，与 contrapz 函数进行比较，有什么异同？哪个函数的精度高？例题中的 contrapz 能否用 consum 代替？

3．axis equal 语句的作用是什么？在什么情况下用到它？

4．请自己编制一个用以绘制莫尔圆的函数 M 文件，并进行调用。重新求解本练习中的问题，对照一下结果。

5．绘图时，hold 命令有什么作用？还有没有实现此功能的其他方法？

练习 46　绘制 Smith 图和波形图

知识背景

从本练习开始，我们来学习 MATLAB 在电学方面的应用。在现代社会，电工与电子技术得到了广泛的应用。从收音机到电子计算机，从医疗诊断器械到飞机、火箭，我们随处可以感觉到电工与电子科学的存在。电工与电子是个统称。具体说起来，基本电路、过渡过程、变压器、电动机、继电器、接触器等属于电工范畴；而半导体二极管、三极管、各种放大电路、数字集成电路、晶闸管应用电路等属于电子范畴。MATLAB 丰富的语句函数和绘图功能以及工具箱为我们更好地研究电工和电子领域的问题提供了可能。

主要内容

【本练习讲述知识点】

本练习作为铺垫，将要介绍电工与电子常见图形在 MATLAB 下的绘制。我们将要使用一些循环语句和嵌套结构来完成特定功能，复习 MATLAB 基本绘图命令（包括坐标选取、坐标轴标注和图线标注）以及傅立叶分析常用函数 fft 的用法。

练习过程

（1）我们首先来学习绘制 Smith 图

凡是从事高频工程领域研究的人，经常会遇到 Smith 图。所谓 Smith 图，是一种反映阻抗与复反射系数之间关系的图形。Smith 图能够清楚地表现出阻抗与复反射系数的联系，具有很强的直观性。因为这些特点，Smith 图得到了广泛应用。

我们设复反射系数为 γ，阻抗为 z(z=r+i*x)，则有：

$$\gamma = (z-1)/(z+1) = u + iv$$

绘制 Smith 图时，我们以 u 为横坐标，v 为纵坐标。设定两个参数 r 和 x，并令其为常量，则我们可以得到一个圆，这就是 Smith 图。那么所得圆的方程是什么呢？

由阻抗和反射系数关系式，我们得到两个实数方程：

$$v^2+(u-\frac{r}{1+r})^2=\frac{1}{(1+r)^2} \quad 和 \quad (v\pm\frac{1}{x})^2+(u-1)^2=\frac{1}{x^2}$$

对于第一个式子，r 为常数，是 x 的轨迹；对于第一个式子，x 为常数，是 r 的轨迹。

如果我们想画 r=0，0.2，0.5，1，1.5，2，4 时的 Smith 图，应该在命令区中输入：

```
plot([0,0],[-1.1,1.1]),hold on,xlabel('u'),
plot([0,0],[-1.1,1.1]),hold on,ylabel('v'),
axis equal,axis([-1.1,1.1,-1.1,1.1]),grid
tr=2*pi*(0:0.005:1);
for r=[0,0.2,0.5,1,1.5,2,4]
rr=1/(1+r);cr=1-rr;
plot(cr+rr*cos(tr),rr*sin(tr))
end
for x=[0,0.2,0.5,1,1.5,2,4]
rx=1/x;cx=rx;
tx=2*atan(x)*(0:0.005:1);
plot(1-rx*sin(tx),cx-rx*cos(tx))
plot(1-rx*sin(tx),-cx+rx*cos(tx))
end
```

按回车后，我们得到图 46-1 所示的图形。

在图中没有标注，读者可以利用 gtext 命令将每条线所代表的意义标注出来。

上面我们利用将复数方程化为两个实数方程的方法，得到了 Smith 图。那么，还有没有别的方法得到 Smith 图呢？将案是肯定的。MATLAB 有着优秀的复数计算功能，我们就用前面给出的复数方程，还是令 r 和 x 中一者为常数。我们来看看具体是怎么实现的。

在命令区内输入：

```
p=logspace(-2,2); x=[-p, p];
q=[0, 0.05, 0.10, 0.20, 0.50, 1, 2, 5, 10];
for r=q
z=(r+i*x-1)./(r+i*x+1);
plot(z), hold on,axis equal
end
r=p;
for x=[-q, q]
z=(r+i*x-1)./(r+i*x+1);
plot(z), hold on, axis equal
end
grid,hold off
```

得到的图形，如图 46-2 所示。

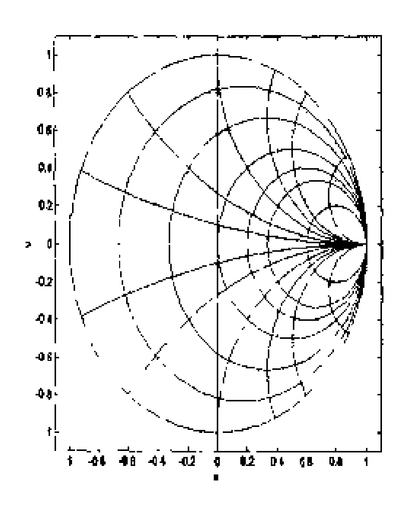


图 46-1 Smith 图

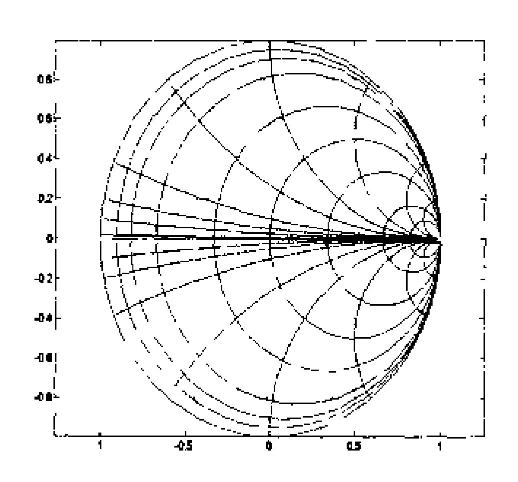


图 46-2 复数法绘制 Smith 图

复数法中最为关键的语句是 z=(r+i*x－1)./(r+i*x+1)，这一句使得 z 与 x 建立了确定的复数域里的函数关系。

（2）绘制波形图

在电子科学中，经常要对波形进行调制，从而得到调制波。一般的调制波方程为：

$y=(A_0+\Delta A)\sin((\omega_0+\Delta\omega)t+\Delta\phi)$

使振幅增量 ΔA，频率增量 $\Delta\omega$ 和相位增量 $\Delta\phi$ 分别随信号变化，就形成调幅、调频和调相。ω_0 称为载频，信号频率一般低于载频，出于方便的考虑，在这里，我们取信号频率为 $0.1\omega_0$。

我们先来绘制调幅波形。

在命令区内输入：

```
clf,format compact,t=0:0.001:1;
A0=10,w0=100,phi0=0;
x=A0.*sin(w0*t+phi0);
subplot(3,1,1),plot(t,x),hold
xlabel('t'),ylabel('载频信号')
dA=5*sin(0.1*w0*t),
dw=0;
dphi=0;
y=(10+dA).*sin((w0+dw).*t+dphi);
subplot(3,1,2),
plot(t,y),ylabel('调制波形')
subplot(3,1,3),
Y=fft(y);plot(abs(Y)),grid
axis([0,50,0,4000]),ylabel('信号频谱')
```

得到的结果如图 46-3 所示。

从图 46-3 可以看到“载频信号”、“调制波形”和“信号频谱”三幅图形，我们明显地

看出，调制波形与载频信号之间的差别。

在程序中，我们用到了 fft 函数命令。fft 函数的作用是离散傅立叶正变换和反变换。傅立叶分析把信号分解成不同频率的正弦函数的叠加。傅立叶变换是信号处理的最重要、最常用的工具之一。

下面我们来绘制调相波形。

仿照上面的方法，在命令区中输入：

```
clf,format compact,t=0:0.001:1;
A0=10,w0=100,phi0=0;
x=A0.*sin(w0*t+phi0);subplot(3,1,1),plot(t,x),hold
xlabel('t'),ylabel('载频信号')
dA=0;
dw=0;
dphi=5*sin(0.1*w0*t);
y=(10+dA).*sin((w0+dw).*t+dphi);
subplot(3,1,2),
plot(t,y),ylabel('调制波形')
subplot(3,1,3),
Y=fft(y);plot(abs(Y)),grid
axis([0,50,0,4000]),ylabel('信号频谱')
```

得到的图形如图 46-4 所示。

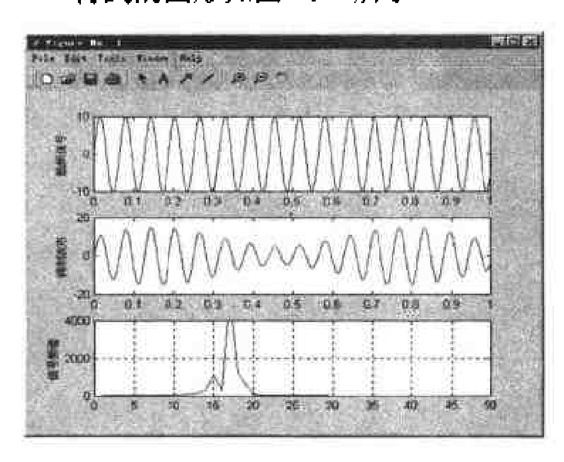


图 46-3 调幅波形

图 46-4 调相波形

比较上面两幅图，差别比较大，但实际上，绘制它们的程序差别非常小。调幅波形中设定增量的语句是：

dA=5*sin(0.1*w0*t)，dw=0；dphi=0；

而调相图形中的设定增量语句是：

dA=0，dw=0；dphi=5*sin(0.1*w0*t)；

可见，要绘制不同调制方式的波形，只需要根据要求设好参数即可。本例中，要得到调幅波形，应该将振幅增量 dA 设为某表达式，而其他两个增量为 0；得到调相图形则是将相移 dphi 设为表达式，其他两项为 0。

【练习小结】

本练习作为本书电工与电子部分的第一个练习，旨在引导读者就如何实现电工电子中常用图形绘制有一个感性认识。我们学习了Smith图的绘制，其中实数方程和复数方程两种方法实现绘图都是应该掌握的内容。不同调制方式产生波形的重点在于函数模型的转化和fft函数的应用。请读者加深理解。

【思考题】

1．绘制Smith图时是如何得到两个实数方程的？
2．绘制 Smith图用到的循环语句结构是什么？还能用别的方法吗？
3．我们如何实现图形中坐标轴和图线的标注？
4．fft函数的用途是什么？
5．请数本练习例题所给数据绘制调频波形。

练习 47　点、线电场计算

知识背景

在现代物理学研究中，电磁场仍旧是一个重要的话题。从安培到法拉第，到麦克斯韦，电磁场的理论一步步得到发展和完善。在实际的学习和研究中，电磁场的理论非常重要。但我们知道，电磁场计算的公式较为繁杂，如果人工计算，就会耗费大量时间。而 MATLAB 为我们提供了简洁、直观的编程语言，使得我们能够通过建立适当的模型，完成数值的计算和反映电磁场特点的图形的绘制。这一点使得 MATLAB 在电磁场方面也有着广泛的用途。对于电磁场的一些公式，我们假定读着都相当熟悉，这里只是简单的加以引用，不作详细的说明。

主要内容

【本练习讲述知识点】

本练习将首先结合求解静电场的实际问题，向大家介绍 MATLAB 在电学方面的应用。我们将介绍着的输出语句 fprintf、spritf 和 disp 以及连用功能。另外通过建立模型，我们将近一步提高模环语句的编程技巧。另外我们还将用到以前学过的 max、min 和 linspace 语句。

练习过程

（1）点电荷电场计算

我们先来学习静电场的计算。首先，我们来着一个具体例子。

已知：平面上有 m 个电荷。求它们之间的库仑引力。

电荷将产生库仑作用力，众多电荷之间将产生相互作用。我们来分析这个问题。

库仑定律的形式是：$F = q_1 q_2 /(4\pi r^2 \varepsilon)$

为了便于计算，我们按 x 和 y 向写出它的分量形式：

$F_x = q_1 q_2 (x_2 - x_1)/(4\pi r^3 \varepsilon)$　　$F_x = q_1 q_2 (x_2 - x_1)/(4\pi r^3 \varepsilon)$

$$r=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

其中 r 为两个电荷之间的距离。

我们的思路是先选定一个电荷，求其他电荷对它的作用力的合力。然后，再选定下一个电荷，进行同样的计算。

请读者仔细琢磨下面的程序。

```
m=input('number=');                                 %输入电荷数目
  for i=1:m                                         %循环语句
      b=input('spot');
      x(i)=b(1);
      y(i)=b(2);
      q(i)=input('Q=');
  end
E=8.85e－12;                                         %常量
c=1/(4*pi*E);
for i=1:m
Fx=0;
Fy=0;
     for
    j=1:m
    if(i~=j)
      xij=x(i)－x(j);yij=y(i)－y(j);                  %计算坐标差值
      rij=sqrt(xij^2+yij^2);
      Fx=Fx+c*q(i)*q(j)*xij/rij^3;                  %计算作用力
      Fy=Fy+c*q(i)*q(j)*yij/rij^3;
    end
end
fprintf('其他电荷的合力=',i)
fprint('Fx=%g', Fx)                                 %输出结果
fprint('Fy=%g',Fy)
```

上面的程序首先要求输入电荷的数目，电荷的坐标及电荷量。对于每一个确定的电荷，求其他电荷对它的作用力的合力，然后以分量的形式表示出来。

这里我们用到了 fprint 命令，这个命令用于把格式化的数据写入文件。它的一般形式是：fprintf（format，A，…）

类似的命令还有 sprintf ，它用来把格式化的数据写入字符串。sprintf 把数循按照要求的格式转化为字符串，再将它与需要显示的字符串组装成一个长字符串，这种方式显得非常方便。%表示数据格式，f 代表浮点数。

比如，我们在命令区里输入：

```
a=sprintf ( '%s', 'hello')
b=sprintf ('The array is %dx%d.',2,3)
```

c=sprintf ('%0.5g',(1+sqrt(5))/2)

将得到如图 47-1 所示的结果。

我们再来看一看 disp 语句。这也是用来输出结果的语句。语句形式如下：

disp（x）

它将显示括号里的内容，可以是字符串，也可以是表达式（以最终结果输出），或者是数值。当括号里是字符串时，必须以单引号括起来。请注意，当括号里是一个没有定义的变量时，系统将认为出错，但如果用引号引起来，将认为是字符串。

为了说明上面的结论，我们来看一个例子。在命令区里输入：

disp（'x'）

disp　34

disp (x)

将得到下面的结果：

x

34

??? Undefined function or variable 'x'.

大家可以很清楚地看到加引号与不加引号之间的区别。对于数值，加不加括号，对于输出结果没有影响。

disp 和 sprintf 可以连用。我们在命令区里输入：

x=0:2:10;

y=[x,x.^2];

disp(sprintf('%10f%10f\n',y))

将得到下面的结果（图 47-2）。



图 47-1　sprintf 语句输出结果

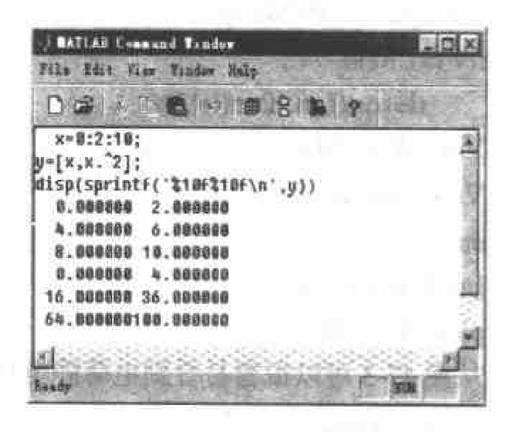


图 47-2　disp 语句和 sprintf 连用

（2）线电荷电场计算

已知：电荷均匀分布在线段（－L，L）上，线电荷密度为 q（C/m）。求平面电位分布。

先建立模型。我们将线电荷分为 M 段，每段长为 dL，每段电荷看作集中在线段中点，

产生的电位为：$dV = \dfrac{qdL}{4\pi r\varepsilon}$　对整段电荷求和。将电荷离原点垂直距离 r 从 0 到 10m 分为 Nr+1 个点。逐点计算电位。取 q=1，L=5，N=50，Nr=50。

在命令区里输入程序段。

得到的计算结果如下：

```
ans =
    1.0e+010 *
    9.31992014169187    0.86540327347324
```

得到的图形如图 47-3 所示。

在程序中，我们用到了 linspace 函数，用来产生等差数列。max 和 min 用来查找结果中的最大值和最小值。这两个知识点，我们前面已经学习过了。这些都是常用的函数，希望读者熟练掌握，灵活运用。

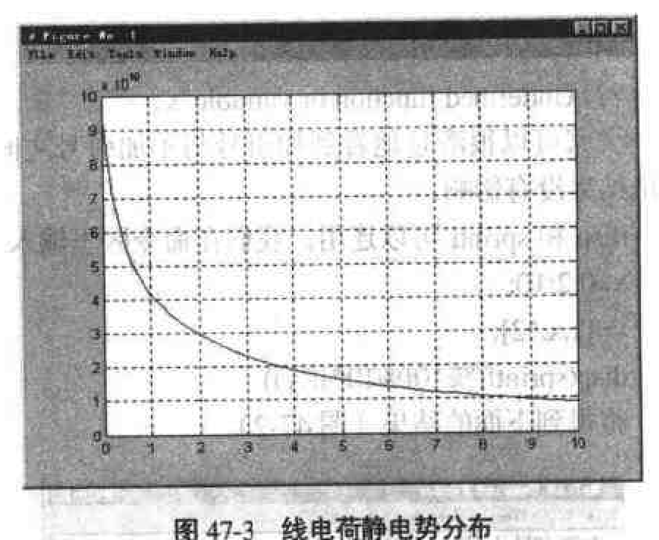
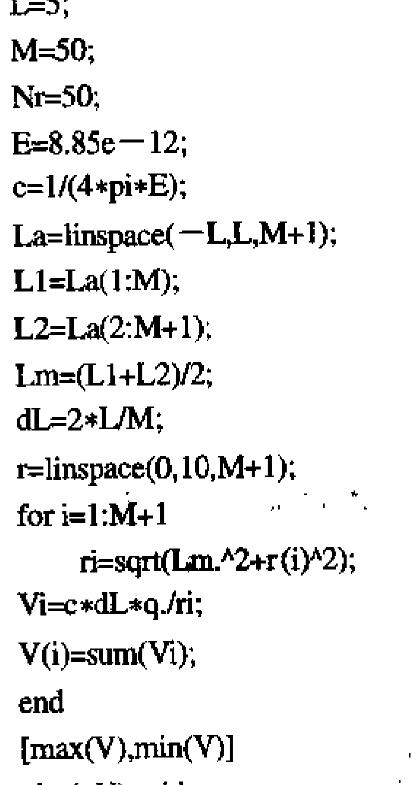


图 47-3　线电荷静电势分布



```
q=1;
L=5;
M=50;
Nr=50;
E=8.85e－12;
c=1/(4*pi*E);
La=linspace(－L,L,M+1);
L1=La(1:M);
L2=La(2:M+1);
Lm=(L1+L2)/2;
dL=2*L/M;
r=linspace(0,10,M+1);
for i=1:M+1
    ri=sqrt(Lm.^2+r(i)^2);
Vi=c*dL*q./ri;
V(i)=sum(Vi);
end
[max(V),min(V)]
plot(r,V),grid
```

从图 47-3 可以很容易看到电荷静电势的分布趋势。

【练习小结】

本练习主要向读者介绍了如何利用 MATLAB 中的编程和数值计算功能来完成对简单电磁场问题的解答。在所举例子中，我们学习了 fprintf 语句和 sprintf 语句的用法。这两种语句都用来完成显示结果的功能。有了它们，我们就能很方便地按照我们喜欢的格式输出运行结果。这两个语句能够将字符串与数值计算的结果联合输出。disp 也用来输出结果。它可以输出字符串和数值。请读者注意 fprintf、sprintf 与 disp 语句连用的功能。它可以在一行里输

出字符串和数值的联合体。我们还复习了 max 和 min 函数及 linspace 语句，希望读者通过复习再仔细体会这些语句函数在科学研究和工程计算中的应用。

【思考题】

1. 在第一个例子中，我们如何实现输入参数的功能？用到了什么语句？
2. 请分析第一个例子中的程序结构，注意双循环语句的执行顺序。
3. fprintf 和 sprintf 语句有什么不同之处？能否互相替换？
4. disp 单独使用和与 fprintf、sprintf 连用，实现的功能有什么不同？
5. 试计算一个均匀带电圆盘在其轴线上某点电场强度的大小。
6. 总结一下解决电学问题的步骤。

练习 48 电场计算

知识背景

我们可以通过不同的方法计算电场，但显然，仅仅得到一个计算结果，我们是不能满意的。MATLAB 有着强大的图形功能，我们就考虑利用这一特点，将我们得到的结果可视化，使数据结论在形象和直观的图形中得到表述。尤其对于电磁场来说，电场、磁场分布，是我们非常关心的方面，如果能将它们清楚地表现在图形中，我们迅速摸清相应电场和磁场的规律，这样做的好处不言而喻。本练习我们就进行这样的尝试。

主要内容

【本练习考查知识点】

本练习首先将向读者介绍如何编程解决电位、电场问题。通过实例，训练读者面对较为复杂的模型，如何仔细分析它的内核，进而编制合理程序解决问题的能力。我们还将学习 MATLAB 中的 eval、feval、clabel 函数语句，并会接触 gradient、quiver 这些命令。通过解决一类问题，使读者能够熟练进行数据可视化处理。

练习过程

绘制等电位线和电场分布图：

如果已知一个电场里的电位分布，我们就能由电位计算电场。

设空间电位分布为：V=V(x,y,z)

由电学知识，可以得到电场等于电位场的负梯度。即：

$$\vec{E} = -gradient(V) = (\frac{dV}{dx}\vec{i} + \frac{dV}{dy}\vec{j} + \frac{dV}{dz}\vec{k})$$

其中，gradient 表示梯度，由一个标量函数产生一个向量函数。i、j、k 分别表示空间的三个方向。下面我们利用 MATLAB 自带的 gradient 函数来建立模型，编制程序。这里应当注意，MATLAB 中的 gradient 是一个数值微分函数，由数学知识我们知道，这种情况下当

选点较密时，计算结果的精度会比较好。对于电位方程式为 V=log(2*x.^2+4*y.^2)

编制的程序如下：

```
V= 'log(2*x.^2+4*y.^2) ';                %以字符串方式输入电位方程式
xmax=10;ymax=5;line=20;                  %确定 x、y 绘图范围和网格线数
xplot=linspace(-xmax,xmax,line);
[x,y]=meshgrid(xplot);                   %生成二维网格
Vplot=eval(V);                           %执行输入的字符串
[Explot,Eyplot]=gradient(-Vplot);        %计算电场
meshc(Vplot);                            %绘制三维曲面
xlabel('x');ylabel('y');zlabel('电位');
axis([-xmax,xmax,-ymax,ymax]);
a=contour(x,y,Vplot)                     %绘制等高线
clabel(a);hold on;
quiver(x,y,Explot,Eyplot);               %绘制 E 箭头图
xlabel('x');ylabel('y');hold off
```

运行后，得到电位图 48-1 和电场分布图 48-2。

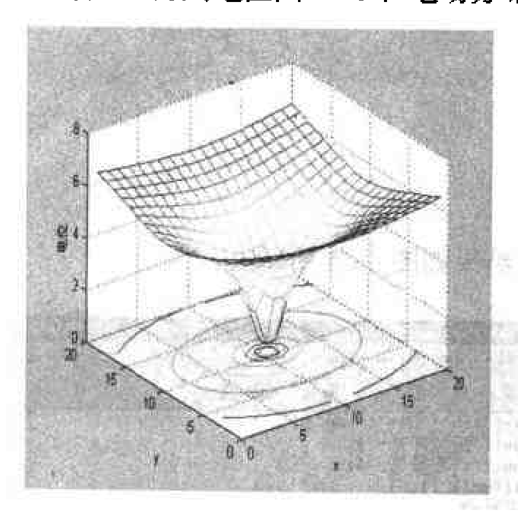


图 48-1　电位三维图

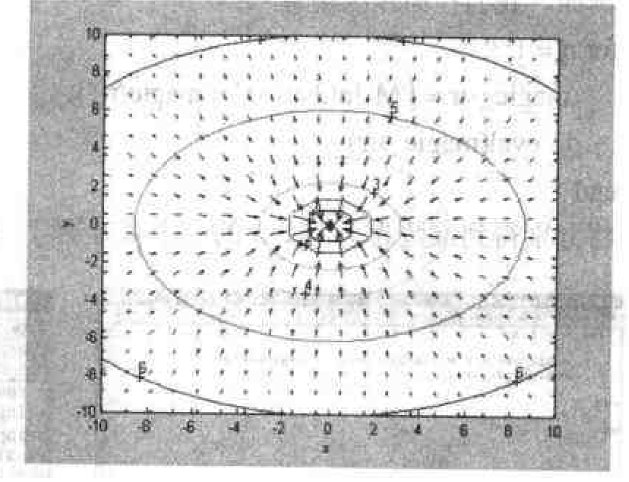


图 48-2　电场分布图

从三维电位图上我们可以清楚地看出电位沿空间分布的趋势。如果 MATLAB 没有这么好的绘图功能，我们将很难建立起电位分布的图形概念。这里我们用到了 gradient 函数，这个函数用来作梯度运算。梯度的概念我们已经介绍过了。有了这个函数，我们将很方便地进行一系列和梯度有关的运算。

再来看一下电场分布图 48-2。

由图我们可以看出，电场方向指向图的中心，箭头清楚地表现出电场的汇聚趋势。另外，图 48-2 还标出了等位线。所有在等位线上的点，它们的电位相等。读者可以从这幅图上很快了解到所给电场分布的信息。

我们还可以再输入一个电位方程式，看看有什么不同。如果输入的电位方程式是：

V= '2*x.^2+4*y.^2 '

将得到图 48-3 和图 48-4。

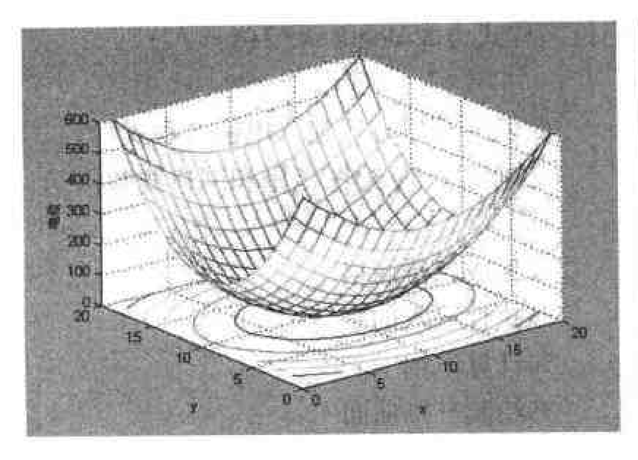


图 48-3　电位图

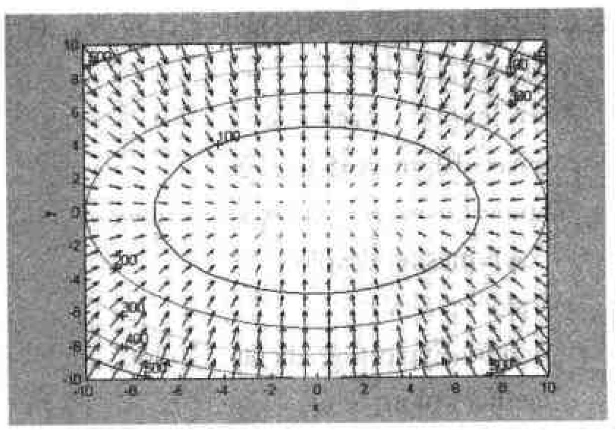


图 48-4　电场分布图

我们可以将这两幅图与图 48-1 和图 48-2 进行比较，可以明显看出电位和电场分布的不同。有了图形功能，对于任何类型的电位和电场分布，我们都能绘出它们的图形，从图形上直观地看出各自的特点，进行比较，得出结论。

在上面的程序中，我们用到了 eval 语句函数。eval 函数的格式为：eval（'字符串'）。

当字符串是 MATLAB 的合法表达式时，它将求得表达式的值。

例如，我们在命令区里输入：

```
for n = 1:3
    magic_str = ['M',int2str(n),' = magic(n)'];
      eval(magic_str)
end
```

将分别得到维数为 1、2、3 的三个魔方矩阵。结果如图 48-5 所示。



图48-5　eval 语句运行结果



图 48-6　feval 函数执行结果

还有一个类似的函数命令 feval。它用来执行字符串代表的文件或函数。比如我们在命令区里输入：

```
fun=['sin';'cos';'log'];
a=input('函数序号=');
b=input('自变量值=');
feval(fun(a,:),b)
```

将得到图 48-6 所示的结果。这个程序首先定义了一个语句函数“fun”，然后要求输入函数序号和自变量的值，最后使用 feval 命令，根据自变量的值计算所选字符串所表示函数的值。

细心的读者应该很快发现，事实上 feval（'cos',x）和 cos（x）是完全等价的。采用 feval 的好处在于能够灵活控制程序执行，达到更好的编程效果。

在绘制电位线的程序中，我们还用到了 clabel 语句。我们知道 countour 语句用来绘制绘制二维等高线图，clabel 语句就是用来标注等高线的。它利用返回的等高线高度值和句柄给那些有足够空间进行标注的等高线进行标注。例如，我们在命令区里输入：

```
[x,y] = meshgrid(-2:.2:2);
z = x.^exp(-x.^2-y.^2);
[C,h] = contour(x,y,z);
clabel(C,h)
```

将得到标注过的图 48-7。

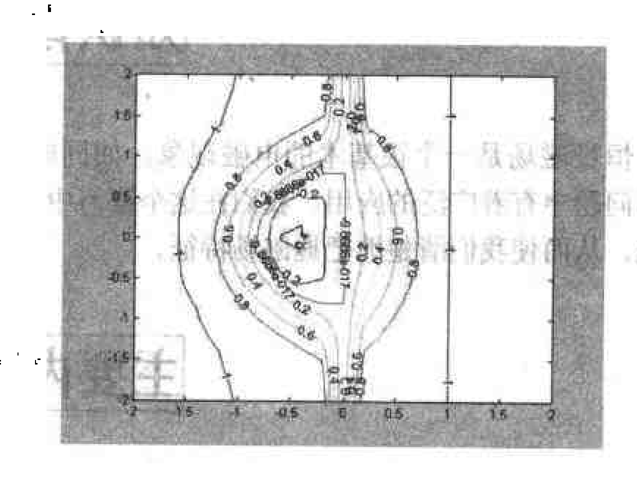


图 48-7　clabel 标注图形

从图 48-7 中，我们看到一幅等高线图。等高线在气象学上有着很重要的应用，我们平常从电视天气预报节目上看到的气压图就是用专用气象软件绘制的。

【练习小结】

本练习主要结合实例向读者介绍了如何编程解决电位、电场问题。利用 MATLAB 中的绘图功能对所得的结果做了可视化的处理。我们重点学习了 eval、feval 这两种编程常用函数语句以及利用返回值对图形进行标注的命令 clabel。另外我们还用到了 gradient 函数，这是用来求梯度的函数，它大大方便了我们对空间问题的求解。

【思考题】

1. 请说出 meshgrid 语句的功能，MATLAB 中还有没有具有类似功能的语句？
2. counter 命令绘制的等高线图是几维的？绘制三维等高线用什么命令？
3. eval 语句和 feval 语句分别应用在什么地方？它们之间有什么练习？
4. gradient 函数的作用对象是什么？梯度运算的结果是标量还是向量？
5. clabel 是如何进行标注的？是不是等高线图上的每根线都会被标注？

练习 49　磁 场 计 算

知识背景

恒稳磁场是一个很基本的电磁现象。如同电场计算一样，磁场计算在科学研究和工程实际问题中有着广泛的应用。我们在这个练习中着手解决磁场问题。并试图用图形将数据可视化，从而使我们清楚地把握磁场特征。

主要内容

【本练习讲述知识点】

本练习考查读者综合使用编程、绘图、逻辑验证等来解决实际磁学问题的能力。我们将利用 linspace 语句、for 循环语句、subplot 和 mesh 绘图命令及逻辑运算符。练习中涉及到较为复杂的程序，希望读者仔细体会。

练习过程

（1）电流环产生的磁场

我们来结合实际例子看一下如何解决这类问题：

我们来看看如何用毕奥－萨伐定律计算电流环产生的磁场。磁学知识告诉我们，载流导线产生的磁场规律为：

任一电流元 $I\vec{dl}$ 在空间任一点 P 处所产生的磁感应强度 $\vec{dB}$ 为：

$$\vec{dB}=\frac{\mu_0}{4\pi}\cdot\frac{I\vec{dl}\times\vec{r}}{r^3}$$

其中，$\vec{r}$ 为电流元到 P 点的矢径，$\vec{dl}$ 为导线圆的长度矢量。则 P 点的总磁场可沿载流导体全长积分各段产生的磁场来求得。

我们在命令区里输入：

```
R=2.5;I0=4;s=4*pi*1e-7;C0=I0*s/(4*pi);
x=linspace(-3,3,20);y=x;
N=20;
t0=linspace(0,2*pi,N+1);
t1=t0(1:N);
y1=R*cos(t1);
z1=R*sin(t1);
t2=t0(2:N+1);
y2=R*cos(t2);
z2=R*sin(t2);
dlx=0;dly=y2-y1;dlz=z2-z1;
xc=0;yc=(y2+y1)/2;zc=(z2+z1)/2;
for i=1:20
    for j=1:20;
rx=x(j)-xc;ry=y(i)-yc;rz=0-zc;
r3=sqrt(rx.^2+ry.^2+rz.^2).^3;
dlXrx=dly.*rz-dlz.*ry;
dlXry=dlz.*rx-dlx.*rz;
Bx(i,j)=sum(C0*dlXrx./r3);
By(i,j)=sum(C0*dlXry./r3);
 end
end
clf;quiver(x,y,Bx,By)
```

得到的结果如图 49-1 所示。

从图中，我们清楚地看出整个电流环的分布情况。

如果我们改变参数，比如将 R 改为 5，看看磁场分布图，如图 49-2 所示。

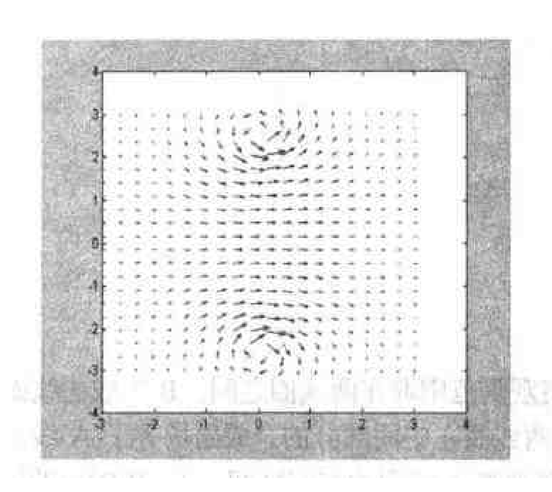


图 49-1 电流环产生的磁场分布

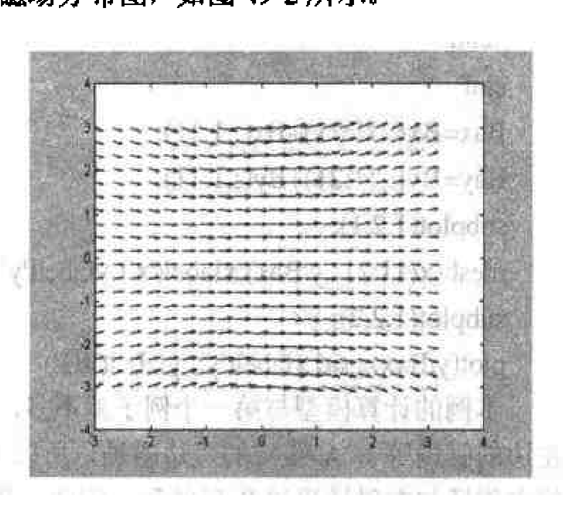


图 49-2 增大 R 后的磁场分布

从图中可以看到，增大 R 后，磁场变化很大。

（2）赫姆霍兹线圈的验证

所谓赫姆霍兹线圈，就是间距正好等于线圈半径的一对相同的共轴载流圆线圈。赫姆霍兹线圈沿轴线附近的磁场的大小十分均匀，而且都沿 x 轴方向。我们来对这个结论进行验证。我们在命令区里输入相应的程序（见下页），运行后，得到图 49-3。

```
R=1;I0=5;s=4*pi*1e-7;C0=I0*s/(4*pi);
NGx=21;NGy=21;
x=linspace(-R,R,NGx);
y=linspace(-R,R,NGy);
N=20;
t0=linspace(0,2*pi,N+1);
t1=t0(1:N);
y1=R*cos(t1);
z1=R*sin(t1);
t2=t0(2:N+1);
y2=R*cos(t2);
z2=R*sin(t2);
dlx=0;dly=y2-y1;dlz=z2-z1;
xc=0;yc=(y2+y1)/2;zc=(z2+z1)/2;
for i=1:NGy
    for j=1:NGx;
rx=x(j)-xc;ry=y(i)-yc;rz=0-zc;
r3=sqrt(rx.^2+ry.^2+rz.^2).^3;
dlXrx=dly.*rz-dlz.*ry;
dlXry=dlz.*rx-dlx.*rz;
Bx(i,j)=sum(C0*dlXrx./r3);
By(i,j)=sum(C0*dlXry./r3);
 end
end
Bax=Bx(:,11:21)+Bx(:,1:11);
Bay=By(:,11:21)+By(:,1:11);
subplot(1,2,1),
mesh(x(11:21),y,Bax);xlabel('x');ylabel('y');
subplot(1,2,2);
plot(y,Bax),grid,xlabel('y');ylabel('Bx')
```

本例的计算模型与第一个例子差不多，但是将观测范围取在两线圈之间。B 生成的线圈左边的磁场等于 A 线圈的左边磁场，所以 A、B 两线圈在中间部分的合成磁场等于 A 线圆的右磁场与左磁场平移 R 后的和。因此，我们需要观测 A 线圈的左右区间 x=[－R,R]内的磁场。

由图 49-3 我们可以得到这样的结论：

在赫姆赫兹线圈的两个线圈之间的轴线附近，有一个很大的区域，x 方向的磁场强度 Bx 比较均匀。类似地，该区域内的 y 方向的磁场强度 By 近似为零。

再来看一下参数设置，如果我们将参数略作改动，即取 R=5，I0=1。运行程序，我们将得到图 49-4。

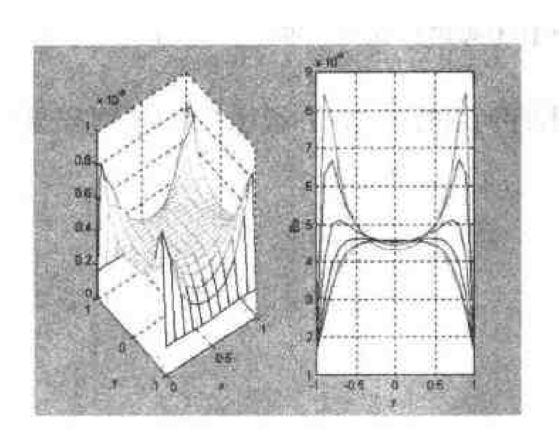


图 49-3　赫姆霍兹线圈附近 Bx 分布

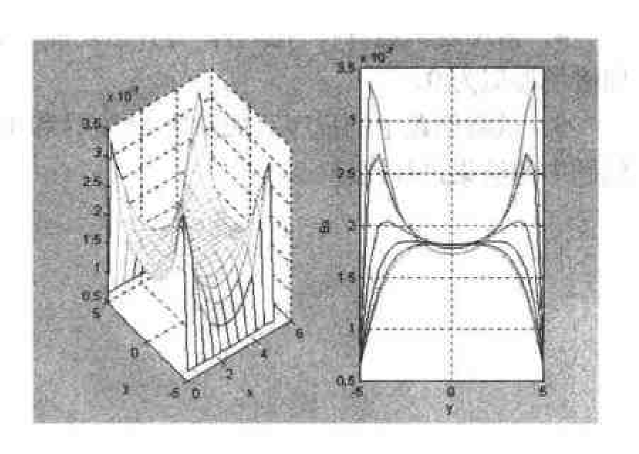


图 49-4　改变参数后的赫姆赫兹线圈 Bx 分布

比较图 49-3 和图 49-4，我们发现图形形状是完全一样的。这说明我们要证明的结论与环半径和电流无关，这也就说明了所证结论的一般性。

我们还可以量化地检验这个结论。因为 Bax（11,6）正好代表磁场中心的磁场强度 Bx，所以我们可以以它为基准，找出相对误差小于 5%的点。

我们在命令区输入：

```
abs((Bax—Bax(11,6))/Bax(11,6))<0.05
```

运行后得到图 49-5 所示的结果。

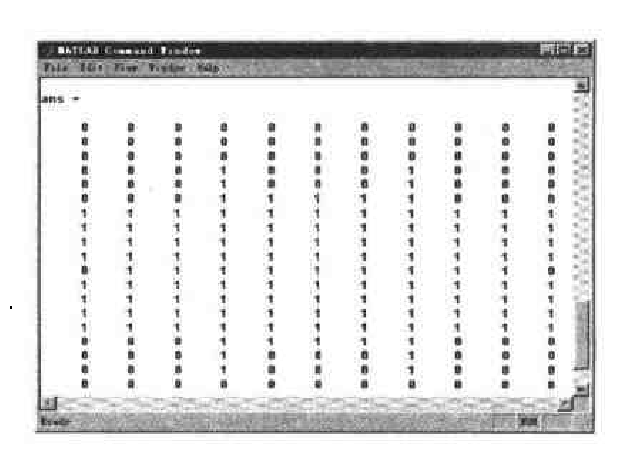


图 49-5　结论量化处理

图 49-5 告诉我们，所有显示为“1”的位置的磁场强度满足相对误差小于 5%，可以看出，结论是相当不错的。

【练习小结】

本练习和上个练习和在一起，解决了几个常见的电磁场问题。本练习中出现了两个较为复杂的程序，都用到了 for 循环即一些绘图命令。相信读者对这部分内容已经相当熟悉了。希望读者能够在理解的基础上进一步研究本练习中的程序。练习的最后，用到了逻辑判断符“<”，符合结论的变量显示为 1，不符合的显示为 0，这样处理使数据一目了然。 请读者注意 MATLAB 中这种处理数据的方法。

【思考题】

1．请思考本练习中两个例子的程序为什么有一部分是完全相同的？

2．subplot 将图形窗口分成若干部分，若不使用这个命令，图形将怎么显示？

3．请用不同方法标注练习中得到的图形。

4．常用逻辑运算符有哪些，有什么功能？试着用逻辑运算符验证第二个例子中 y 轴方向磁场近似为 0。

5．试将例题中的程序写成函数 M 文件和主程序两用的形式，并运行，将得到的结果与例题中的结果进行比较。

练习 50　晶体管放大电路

知识背景

集成电路是六十年代发展起来的一种新型电子器件。它是在半导体硅片上通过一系列工艺制出晶体三极管、电阻、电容及相互间的连线，构成一个完整的有一定功能的电路。运算放大器是线型集成电路的一种，是从最初用于模模电子计算机作为直流电压运算部件而发展起来的。由于集成运算放大器有良好的性能，目前被广泛应用在计算技术、自动控制、无线电技术和各种电与非电量的电测线路中。

主要内容

【本练习讲述知识点】

本练习我们涉及到了电子技术里非常重要的运算放大器。这部分内容希模读者查阅相关书籍，先大致作一个了解。在例题中，我们用到了 poly、polyval 语句，在绘图时，我们用 semilogx 和 gtext 进行处理。最后向读者介绍了 semilogy 函数。本练习语句较多，希望大家好好体会。

练习过程

运算放大器：

我们首先来回顾有关运算放大器的一些概念。运算放大器（简称运放）是一个直接模合的多极放大器，其内部电路通常由差动输入极（用以提高输入电阻和减小零点漂移），中间放大极和射极输出器（用来提高带负载能力）组成。

经过二十多年的发展，目前我们国家已经生产出第四代集成运放产品。这些运放从性能上已达到了比较好的程度。

我们来看下面的例子：

有一个运算放大器电路如图 50-1 所示，试分析放大器开环增益和频率响应对整个电路闭环频率响应的影响。

我们来分析这个问题。设运算放大器的开环增益为 A，它是频率的函放，在图示的连

接方法下，闭环输出与输入电压的比为：

$$H=\frac{U_0}{U_i}=-\frac{Z_2/Z_1}{1+(1+Z_2/Z_1)/A}$$

如果增益 A 很大，分母可近似认为等于 1，进而得到理想运放的闭环传递函放

$$H(s)=\frac{U_0(s)}{U_i(s)}=\frac{Z_2(s)}{Z_1(s)}$$

上式中的 s 为拉普拉斯算子，如果将它换成 jω，放可以得出频率响应，所以这两个式子都是复放方程。问题的实质是考虑 A=A(ω)对 H（ω）的影响。

一般情况下，运算放大器的开环传递函数中包括 3 个实极点，即

$$A(s)=\frac{A_0}{(1+\frac{s}{\omega_1})(1+\frac{s}{\omega_2})(1+\frac{s}{\omega_3})}=\frac{A_0\omega_1\omega_2\omega_3}{(s+\omega_1)(s+\omega_2)(s+\omega_3)}=\frac{b}{a(s)}$$

其中，$\omega_1<\omega_2<\omega_3$，取符号后为其 3 个极点。$A_0$ 为直流情益。

为了避免产生自激现象，放常使 ω_1 和 ω_2 差得很大，例如 $\omega_1<500$ 1/s，$\omega_2>10^6$ 1/s，同时 ω_2 和 ω_3 也应该有一定差距。我们设 $\omega_1=500$，$\omega_2=2\times10^6$，$\omega_3=5\times10^7$，设 $Z_1=2\text{k}\Omega$，Z_3 分别取 $20\text{k}\Omega$、$100\text{k}\Omega$ 和 $500\text{k}\Omega$，我们来求 H（ω）。

考虑到运算放大器放性所可以由图形清楚地表示出来，我们试着用图形来表示情果。如果产生自激现象，将在图像上清楚地反映出来，这样我们情输从整体上把握运算放大器的性质。我们在命令区里输入：

```
Z2=[20,100,500]*1000;
Z1=2000;
A0=2e6;w1=500;w2=2e6;w3=5e7;
w=logspace(2,8);
b=A0*w1*w2*w3;
a=poly([-w1,-w2,-w3]);
A=polyval(b,j*w)./polyval(a,j*w);
for i=1:3
Z12(i)=Z2(i)/Z1;
H(i,:)=-Z12(i)./(1+1+Z12(i)./A);
semilogx(w,abs(H(i,:))),hold on
end
v=axis;axis(v);
semilogx(w,abs(A))
hold off
gtext('Z2=500k')
gtext('Z2=100k')
```

```
gtext('Z2=20k')
gtext('开环增益')
gtext('角频率')
gtext('增益 abs(H)')
```

运行后，得到图 50-2。

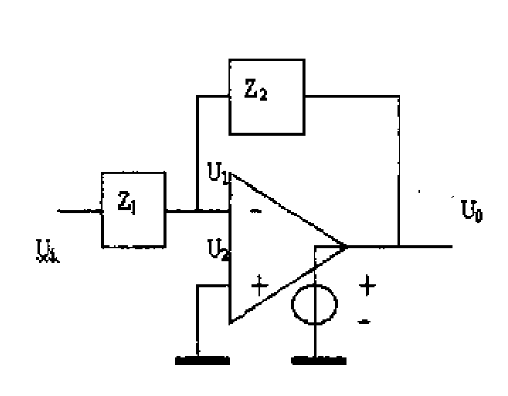


图 50-1　运算放大器电路

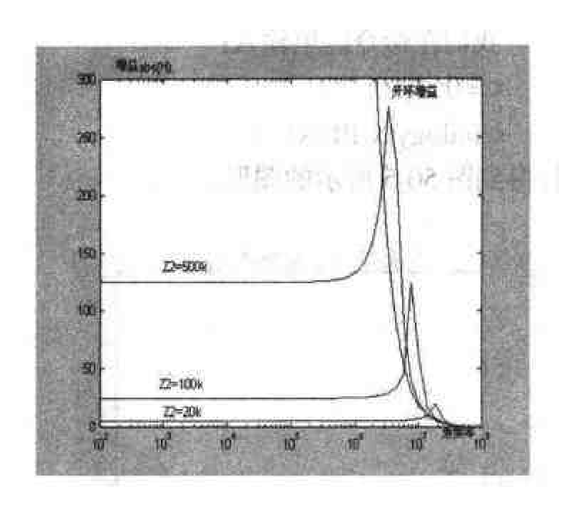


图 50-2　运算放大器闭环频率响应

由图可以看出，运放在低频区较宽的一个频带里具有平坦的增益 $\frac{Z_2}{Z_1}$ 。在高频区里出现了谐振峰，这容易造成运算放大器的自激现象。（自激振荡是这样一种现象，当一个电路满足一定条件后不需要输入信号却能够产生幅度与频率稳定的输出信号。这种现象发生时称电路产生了自激振荡。）

消除自激的方法，可以是减小 ω_1 ，或者增加 ω_2 和 ω_3 。圆盘由于 ω_2 和 ω_3 由运算放大器本身性能决定。因此，在放大器已经选定的情况下，通常只能用加消振电容的方法减小 ω_1 。我们将例子中的 ω_1 由 500 减小为 50，将得到如图 50-3 所示的频率响应图。

从图上我们可以看到，自激的可能性大大减小。如果再减小一点，将 ω_1 减小为 25，将得到如图 50-4 所示的图形。

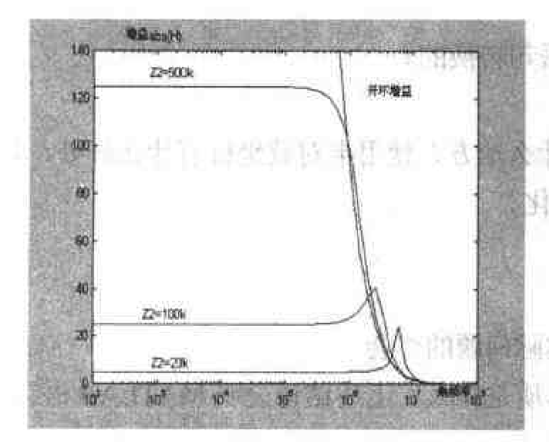


图 50-3　频率响应

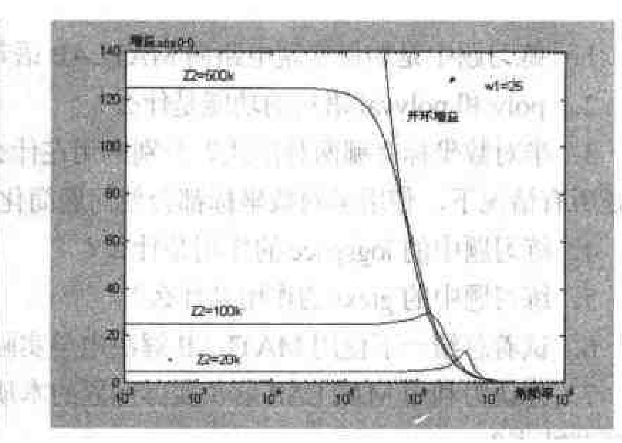


图 50-4　闭环频率响应

可以看到，随着 ω_1 的减小，自激现象出现的几率越来越小。

在例题中，我们用到了 semilogx 绘图命令，它的常用格式是：semilogx（x，y）。这是半对数坐标，x 轴为常用对数坐标。

事实上，还有一个半对数坐标命令，即：semilogy（x，y）。它以 y 轴为常用对数坐标。

我们在命令区里输入：

```
x = 0:.1:10;
semilogy(x,10.^x)
```

将得到图 50-5 所示的图形。

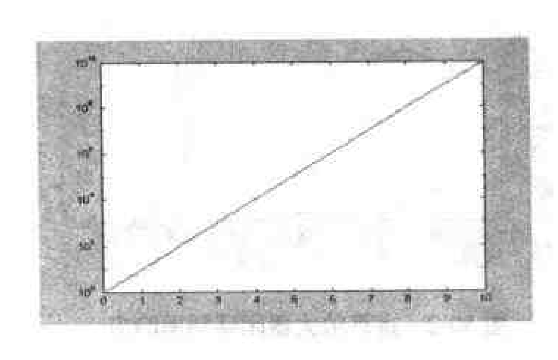


图 50-5　半对数座标

从图中 50-4 中，我们看到在不同阻抗下开环增益变化的规律。与图 50-3 相比，ω_1 减小了，相应地，产生自激的可能性也减小了。

图 50-5 中，我们采用了半对数坐标，图形为一条直线。如果不用半对数坐标，图形将变的复杂，使我们一时难以掌握数据之间的联系和规律。合理使用特殊坐标将大大简化图形输出结果，增强表达效果。

【练习小结】

本练习主要向读者介绍了利用 MATLAB 语句研究运算放大器的方法。对于一个电子器件进行模拟研究，建立合理模型是关键。本练习对读者的编程能力是一个考验，对于复杂问题建立模型，选择表达语句，程序调试，绘制图形，是现代研究常用的方法。semilogx 和 semilogy 都是半对数坐标，有时候，用半对数坐标绘图，会使图形清晰、简洁。也可以将两个命令连用，使两条轴都采用半对数坐标。

【思考题】

1．练习题中是如何实现电路向 MATLAB 语句转换的？

2．poly 和 polyval 语句的功能是什么？

3．半对数坐标有哪两种形式？分别应用在什么地方？使用半对数坐标有什么好处？是不是所有情况下，使用半对数坐标都会使问题简化？

4．练习题中的 logspace 的作用是什么？

5．练习题中的 gtext 的作用是什么？

6．试着总结一下使用 MATLAB 解决电学实际问题的体会。

7．你认为利用 MATLAB 解决实际问题的本质是什么？上述结论说明 MATLAB 语言具有哪些特点？